Range Commanders Council

TELEMETRY GROUP

IRIG STANDARD 107-98

DIGITAL DATA ACQUISITION AND ON-BOARD RECORDING STANDARD

WHITE SANDS MISSILE RANGE
KWAJALEIN MISSILE RANGE
YUMA PROVING GROUND
DUGWAY PROVING GROUND
ABERDEEN TEST CENTER
NATIONAL TRAINING CENTER

ATLANTIC FLEET WEAPONS TRAINING FACILITY
NAVAL AIR WARFARE CENTER WEAPONS DIVISION
NAVAL AIR WARFARE CENTER AIRCRAFT DIVISION
NAVAL UNDERSEA WARFARE CENTER DIVISION, NEWPORT
PACIFIC MISSILE RANGE FACILITY
NAVAL UNDERSEA WARFARE CENTER DIVISION, KEYPORT

30TH SPACE WING
45TH SPACE WING
AIR FORCE FLIGHT TEST CENTER
AIR FORCE DEVELOPMENT TEST CENTER
AIR WARFARE CENTER
ARNOLD ENGINEERING DEVELOPMENT CENTER
GOLDWATER RANGE
UTAH TEST AND TRAINING RANGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
JUNE 1998

3. REPORT TYPE AND DATES COVERED

4. TITLE AND SUBTITLE
DIGITAL DATA ACQUISITION AND ON-BOARD RECORDING STANDARD

5. FUNDING NUMBERS

6. AUTHOR(S)

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
TELEMETRY GROUP
RANGE COMMANDERS COUNCIL
WHITE SANDS MISSILE RANGE, NM 88002-5110

8. PERFORMING ORGANIZATION REPORT NUMBER
IRIG STANDARD 107-98

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
STEWS-RCC
RANGE COMMANDERS COUNCIL
WHITE SANDS MISSILE RANGE, NM 88002-5110

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
SAME AS BLOCK 8

11. SUPPLEMENTARY NOTES
NEW DOCUMENT

12a. DISTRIBUTION / AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This document is a technical standard for use in developing packetized on-board recording systems and has been prepared by the Range Commanders Council (RCC) Telemetry Group (TG). The packet telemetry concept described herein is the baseline for on-board recording of missions that require cross support between organizations. This standard establishes a common framework and provides a common basis for the data structures of on-board recording data.

14. SUBJECT TERMS
telemetry, packetized on-board recording systems

15. NUMBER OF PAGES
83

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
unclassified	unclassified	unclassified	none

IRIG STANDARD 107-98

DIGITAL DATA ACQUISITION AND ON-BOARD RECORDING STANDARD

JUNE 1998

Prepared by

TELEMETRY GROUP
RANGE COMMANDERS COUNCIL

Published by

Secretariat
Range Commanders Council
U.S. Army White Sands Missile Range,
New Mexico 88002-5110

TABLE OF CONTENTS

FIGURES **PAGE**

REFERENCES

[1] "Packet Telemetry," Blue Book, CCSDS 102.0-B-4, November 1995.
[2] "Time Code Formats," Blue Book, CCSDS 301.0-B-2, April 90.
[3] "Telemetry Channel Coding," Blue Book, CCSDS 101.0-B-3, May 93.
[4] "Telemetry: Concept and Rationale," Green Book, CCSDS 100.0-G-1, December 87.
[5] "Report on Digital data Acquisition Standard," Rothrock, M.E. and Rossetti, A.M., Prepared for Telemetry Group of the Range Commanders Council, July 18, 1996.
[6] "Reference Model of Open Systems Interconnection," International Organization for Sization, Draft International Standard DIS-7498, February 1982 or later issue.
[7] "Telemetry Standards," Range Commanders Council Telemetry Group, IRIG Standard 106-96, May 1996.

DOCUMENT CONTROL

A. FIRST ISSUE	
DOCUMENT REFERENCE:	IRIG 107-98
TITLE:	Digital Data Acquisition and On-Board Recording Standard; Packet Telemetry Issue 1
DATE:	1998

STATEMENT OF INTENT

A large number of unique data structures have been developed lately for specific data recording applications that require unique decoding software programs. Writing unique decoding software, checking the software for accuracy, and decoding the data tapes is extremely time consuming and costly.

Therefore, the need exists for a digital data acquisition standard for digital data recording that supports the multiplexing of multiple data streams and maintains the accuracy of data correlation with time. Specifically, this digital data acquisition standard should be compatible with the multiplexing of both synchronous and asynchronous digital inputs such as Pulse Code Modulation (PCM) and MIL-STD-1553 asynchronous data bus, digital voice, time, discrete, and RS-232/422 communication data. In addition, the new standard should be aligned with current developments in layered communications architecture.

This digital data acquisition standard should allow the use of a common set of playback/data reduction software, take advantage of emerging random access recording media, provide an high efficiency multiplexing technique, and take advantage of the rapid improvement in commercial communication technology.

Packet telemetry represents an evolutionary step from the traditional Time-Division Multiplex (TDM) method of acquiring, recording, and playing back scientific applications and engineering data from instrumented vehicle sources to ground data systems sinks. It also relies on a layered architectural model to isolate independent interfaces. The packet telemetry process has the conceptual attributes of:

1. Defining a logical interface and protocol between an instrument and its associated on-board recorder/playback and ground support equipment which remains constant throughout the life cycle of the instrument (bench test, integration, flight, and possible re-use).
2. Simplifying overall system design by allowing microprocessor-based symmetric design of the instrument control and data paths compatible with commercially available components and interconnection protocol standards.
3. Facilitating interoperability of instrumented vehicle systems whose data acquisition and on-board recording systems interfaces conform to IRIG guidelines.
4. Enabling the delivery of high-quality data products in a mode, which is faster and less expensive than would be possible with conventional methods.

The Consultive Committee on Space Data Systems (CCSDS) Packet Telemetry Recommendation is intended primarily to support multi-point to multi-point data transfer over space based transmission links. This standard is primarily to support point to point data acquisition and recording and subsequent playback on a ground based data reduction system. While there are significant similarities, the differences required some minor deviations from the absolute adoption of the CCSDS Packet Telemetry Recommendation. It is therefore the intent of this standard to duplicate the CCSDS Packet Telemetry Recommendation deviating only where absolutely necessary to accommodate on-board recording/ground playback or point to point

considerations. The structure of the CCSDS Packet Telemetry Recommendation, both source packet and transfer frame, are unchanged. This minimum deviation from an existing international standard will allow for maximum use of existing and proposed CCSDS compatible hardware and software.

FOREWORD

This document is a technical standard for use in developing packetized on-board recording systems and has been prepared by the Range Commanders Council (RCC) Telemetry Group (TG). The packet telemetry concept described herein is the baseline for on-board recording of missions that require cross support between organizations. This standard establishes a common framework and provides a common basis for the data structures of on-board recording data.

This standard is an adaptation of the Consultative Committee on Space Data Systems (CCSDS) Packet Telemetry Recommendation contained in reference [1].

> The CCSDS Packet Telemetry Recommendation is limited to the telemetry formats which are generated by the vehicle while the channel coding and synchronization mechanisms required to implement over-the-air transmission of acceptable quality are defined in reference [3].Since this standard is for on-board recording versus over-the-air transmission, most of reference [3] is not directly applicable.
>
> The CCSDS Packet Telemetry Recommendation also includes references to specific time code formats as defined in reference [3] and this standard adopts a portion of reference [3].
>
> The CCSDS Packet Telemetry Recommendation incorporates a concept and rationale document, including "Application Notes," which is contained in reference [4].

For ease of use this standard has been prepared as a stand-alone document and contains only appropriate wording changes to the adopted CCSDS Packet Telemetry Recommendation of reference [1]. The data structure for both the source packet and transfer frame remains unchanged.

> The concept and rationale for this standard, including "Application Notes" are contained in appendix A.
>
> The specific portions of reference [3] appropriate to on-board recording channel coding are included in appendix B of this standard. The specific portions of reference [2] appropriate to time code formats are included in appendix C of this standard.

1. INTRODUCTION

1.1 PURPOSE

The purpose of this document is to establish a common standard for the implementation of digital data acquisition and on-board recording "packet telemetry" systems by the organizations participating in the Range Commanders Council (RCC). This standard is an adoption of the Consultative Committee on Space Data Systems (CCSDS) Packet Telemetry Recommendation (see Reference [1]).

1.2 SCOPE

Packet telemetry is a concept that facilitates the transmission of vehicle-acquired data from source to user in a standardized manner. Packet telemetry provides a mechanism for implementing common data transport structures and protocols that may enhance the development and operation of mission systems.

This standard addresses end-to-end transport of mission data sets from source application processes located on a vehicle to an on-board recording device and then played back to user application processes located on the ground.

This standard is limited to describing the recording formats which are generated by the vehicle in order to execute its role in the above processes.

An overview of the packet telemetry concept is given in Chapter 2.

1.3 APPLICABILITY

The CCSDS Standard includes comprehensive specification of the structure of data streams that are generated for recording by on-board recorders and then played back through mission data processing facilities. The standard does not attempt to define the architecture or configuration of these data processing facilities, except to describe assumed ground data handling services which affect the selection of on-board formatting options.

The CCSDS Standard specifies a wide range of formatting capabilities which may facilitate a high degree of flexibility in the design of on-board data acquisition systems; however, compatibility with the packet telemetry concept may be realized by only implementing a narrow subset of these capabilities. Application notes for implementation of DDARS is included in appendix A.

1.4 RATIONALE

The CCSDS and the RCC-TG believes it is important to document the rationale underlying the recommendations chosen so that future evaluations of proposed changes or improvements will not lose sight of previous decisions. The concept and rationale for CCSDS packet telemetry may

be found in reference [4] and the concept and rationale for the RCC Digital Data Acquisition and On-Board Recording Standard may be found in appendix A.

1.5 STRUCTURE OF THE DOCUMENT

For the designation of text partitions the following conventions will be used.

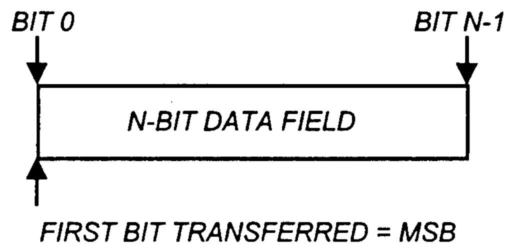

Figure 1-1. Bit Numbering Convention

Text designated by one number belongs to a chapter.
Text designated by two numbers belongs to a section.
Text designated by three numbers belongs to a sub-section.
Text designated by four numbers belongs to a paragraph.
Text designated by a lower case letter belongs to an item.

All specifications are contained in Chapters 3, 4 and 5 of this standard. They are identified by an item number consisting of the number of the text partition as defined above, and a lower case letter. The conventions and definitions applied in these specifications are itemized in section 1.6.

All other text and all figures in these chapters represent comments to these specifications. All comments are printed in Italics. The contents of the specifications take precedence over those of the comments. All major terms used herein are referenced in the index.

1.6 CONVENTIONS AND DEFINITIONS

The following items contain the conventions, which have been used throughout this standard.

a. To identify each bit in an N-bit field the first bit in the field to be transferred (i.e., the most left justified when drawing a figure) is defined to be "Bit 0"; the following bit is defined to be "Bit 1" and so on up to "Bit N-1." When the field is used to express a binary value (such as a counter), the most significant bit shall be the first bit of the field, i.e., "Bit 0" (see Figure 1-1).
b. In accordance with modern data communication practice, vehicle data fields are often grouped into 8-bit words, which conform, to convention 1.6.a. Throughout this standard, such an 8-bit word is termed an "octet."
c. The numbering for octets within a data structure starts with 0.
d. The term "mission phase" designates a period of a mission during which specified on-board recording characteristics are fixed. The transition between two consecutive mission phases may cause an interruption of the on-board recording services.
e. Certain characteristics of the data structures specified in this standard are required to remain unchanged throughout a mission phase or throughout all mission phases. In these cases the term "static" is used to specify characteristics which remain unchanged either with

respect to an application process identifier (for definition see paragraph 3.1.2.3), or within a specific virtual channel (for definition see item 5.e) or within a specific master channel (for definition see item 5.d).

f. Idle data is data that carries no information, but is sent to meet timing or synchronization requirements. The bit pattern of idle data is not specified.

2. OVERVIEW

This packet telemetry standard describes data structures used to transport data from data sources on board a vehicle to an on-board recording device, then played back through ground data systems to data sinks on the ground, as shown in Figure 2-1.

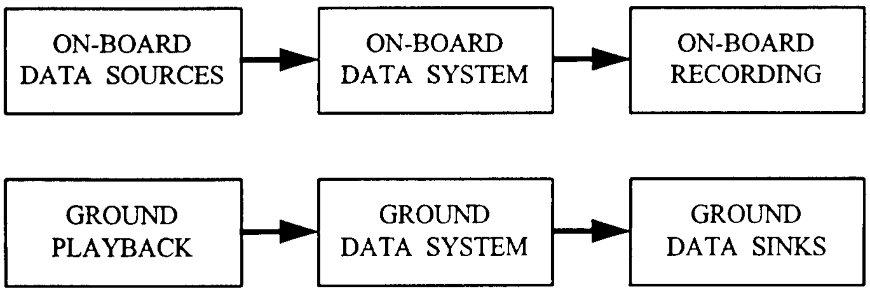

Figure 2-1. RCC Packet Telemetry Data System

2.1 PACKET TELEMETRY CONCEPT

The essence of the packet telemetry concept is to permit multiple application processes running in on-board sources to create units of data (packets) as best suits each data source. The concept permits the on-board data system to record these packets in a way that enables the ground playback system to recover the individual packets with high reliability and to provide them to the data sinks in sequence. These on-board sources are either instruments or sub-systems.

To accomplish these functions, this standard defines two data structures: source packets and transfer frame and a multiplexing process to interleave source packets from various application processes into transfer frames.

2.2 SOURCE PACKET

The source packet, which in the following text may also be termed "packet," is a data structure generated by an on-board application process in a manner responsive to the needs of that process. It can be generated at fixed or variable intervals and may be fixed or variable in length. Aside from a packet header that identifies the source and characteristics of the packet, the internal data content of the source packet is completely under the control of the application process.

The source packet allows each application process within a data source to optimize the size and structure of its data set with a minimum of constraints imposed by the on-board digital data recording system. Each data source is independent of other data sources and can adapt its data structure to the various modes of the instrument or sub-system.

The source packet primary header contains an application process identifier used to route the packet to its destination sink. The header also carries information about the length, sequence, and other characteristics of the packet. An optional source packet secondary header is provided for standardized time tagging of source packets, and to carry application-unique ancillary data.

2.3 TRANSFER FRAME

The transfer frame is a data structure that provides an envelope for recording packetized data. It carries information in the transfer frame primary header that permits the ground system to route the transfer frames to their intended destination. The transfer frame is of fixed length (for a given physical channel during a mission phase).

> *The physical channel is the single bit stream that is recorded onto the physical media. It includes all multiplexed transfer frames as well as any implementor-unique coding algorithms.*

Multiple, individual, asynchronous application processes on board a vehicle can generate variable-length source packets at different rates, and these source packets can then be multiplexed together into fixed-length coded transfer frames.

The transfer frame primary header provides the necessary elements to allow the variable-length source packets from a number of application processes on a vehicle to be multiplexed into a sequence of fixed-length frames. Short packets may be contained in a single frame, while longer ones may span two or more frames. Since a packet can begin or end at any place in a frame, the entire data field of every frame can be used to carry data; there is no need to tune the sizes of packets or their order of occurrence to fit the frames.

A mechanism (idle packets) is provided for cases where a frame must be released and insufficient packet data is available. Further, frames containing idle data are defined to keep the data capture element in synchronization in the absence of data, if required.

On the ground, the information in the frame and packet headers allows the data acquisition system to extract packets in a standardized way.

In addition to packets, the transfer frame can carry one optional field, the transfer frame secondary header. The transfer frame secondary header can be used to carry fixed-length mission specific data. *[Use of privately defined data is not supported by this standard.]*

2.4 SHARING TRANSMISSION RESOURCES

As most DOD recording and communication systems are capacity-limited, multiple data channels must share access to the recorded physical channel. Therefore, the on-board data system must be able to manage the data flow to the recording device in an orderly manner. In addition, different types of data may be handled differently on the vehicle or during playback on

the ground. This standard provides the method of virtual channelization for controlling the data flow.

Virtual channelization is a mechanism that allows the various sources which generate packets to be "virtually" given exclusive access to this physical channel by assigning them recording/transmission capacity on a frame-by-frame basis. Each transfer frame is identified as belonging to one of the up to eight virtual channels. Virtual channelization is normally used to separate sources or destinations with different characteristics. For example, if a vehicle contains an imaging instrument which produces packets containing many thousands of octets, and a number of other instruments which generate smaller packets, a possible system architecture would be to assign the imaging instrument packets to one virtual channel and to handle the rest by multiplexing them onto a second virtual channel. Virtual channels may also be used to allow easy separation on the ground of data streams that are to be sent to different destinations.

Figure 2-2 shows an example of the flow of on-board recorded data from several on-board sources (instruments or sub-systems), through to the playback of the same data to sink processes on the ground. At the top of the figure, generation of source packets from application processes in several data sources is shown. These packets are multiplexed into the transfer frames of several virtual channels. These transfer frames are recorded on-board, using appropriate error protection and synchronization techniques. On the ground, during playback, they are demultiplexed into virtual channels, and the packets are extracted. Source packets are then delivered to sink processes, shown at the bottom of the figure, using the application process identifiers in the source packet headers for routing. Source packets with a given application process identifier may be delivered to one or more sink processes. Packets may be time-ordered prior to delivery using the information in the packet primary header and the packet secondary header.

An example of the implementation of a typical on-board recording data flow is also shown in Figure 2. *Additional details of implementation options are contained in appendix A.*

Figure 2-3 shows a reference model and the extent of coverage of this standard. Application processes have separately defined RCC/IRIG standards. At the coding layer, the coding and synchronization, if required, are not specified.

2.5 APPLICATION NOTES

Application Notes that describe how compatibility with these various data structures may be achieved are presented in reference [4] along with key elements of the rationale behind packet telemetry. Application notes specific to the Digital Data Acquisition and On-Board Recording Standard are presented in appendix A.

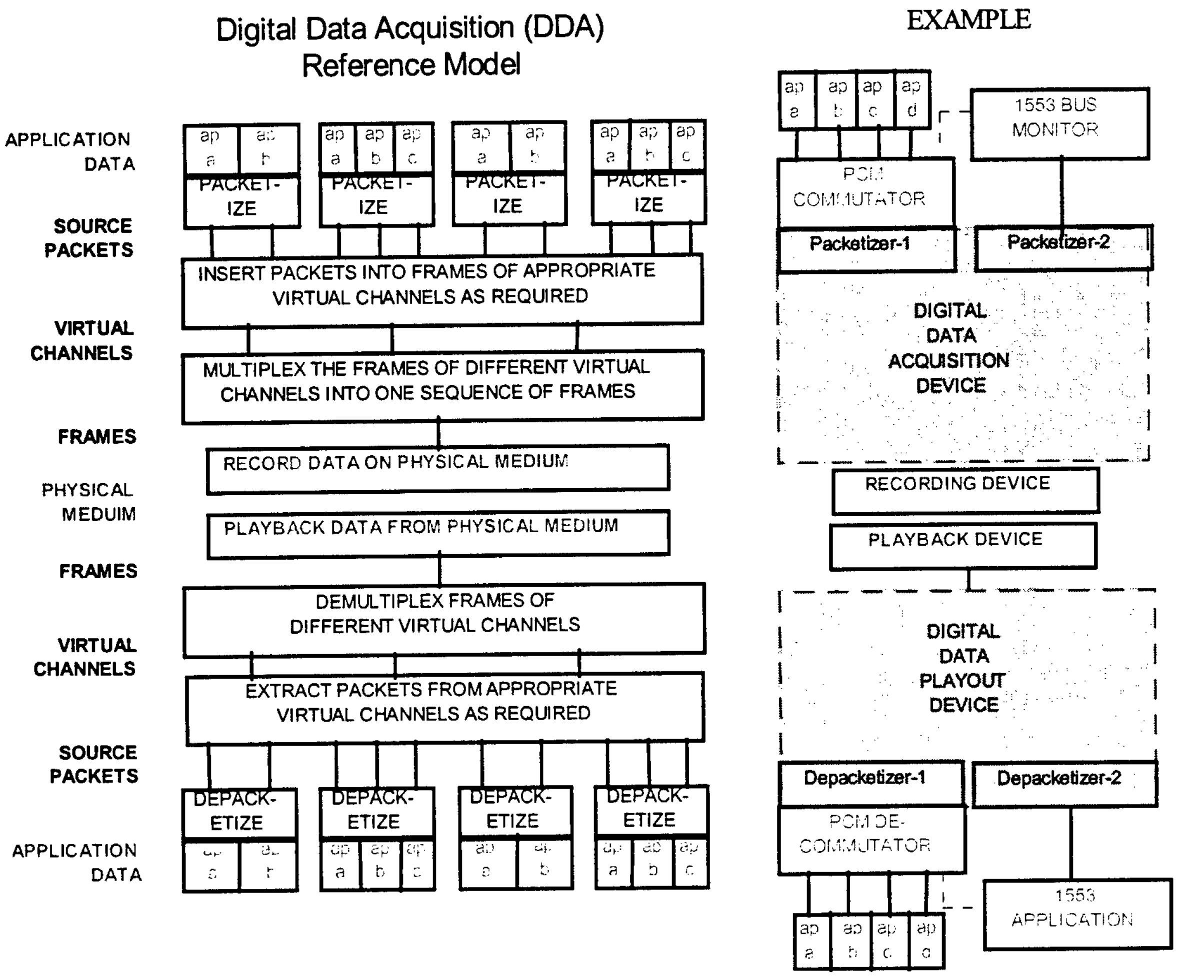

Figure 2-2. Example of on-board recorded data flow.

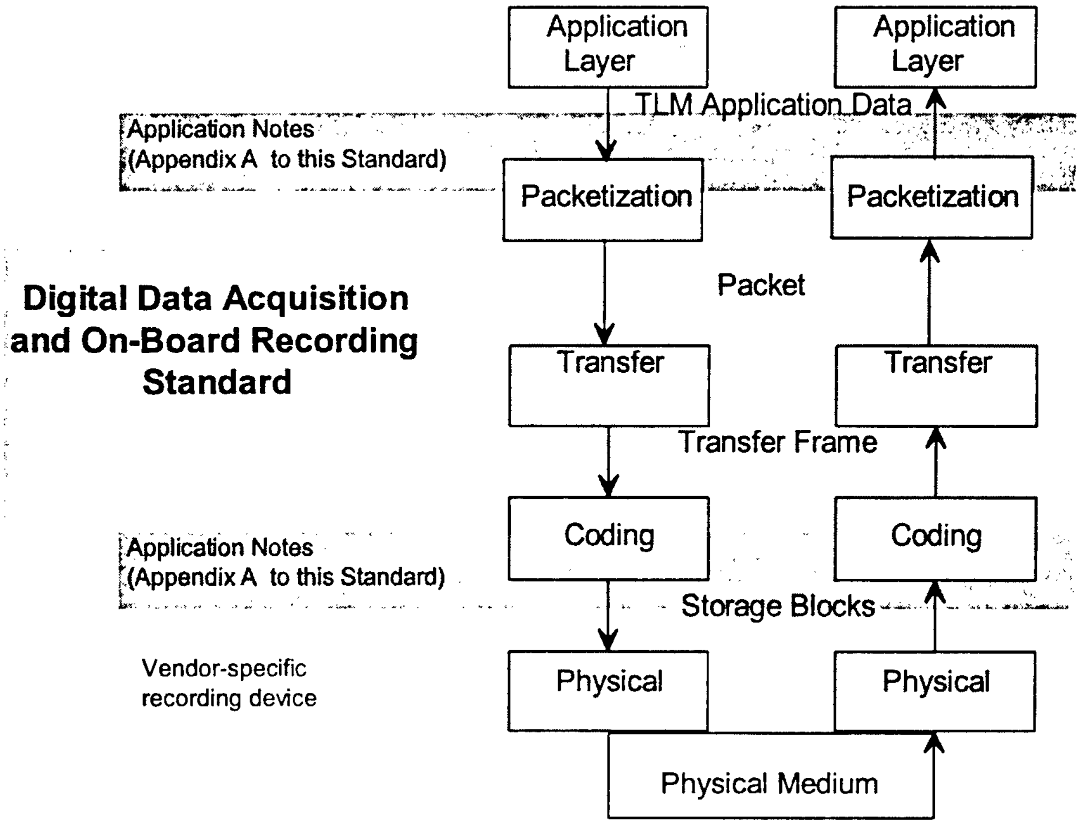

Figure 2-3. Digital data acquisition and on-board recording standard reference model.

3. SOURCE PACKET

a. A source packet, which in the following text may also be termed packet, shall encapsulate a block of observational and ancillary application data which is to be recorded from an application process on-board a vehicle; then played back to one or several sink processes on the ground.

b. The source packet shall consist of two major fields, positioned contiguously, in the following sequence:

	Length in bits
Packet primary header (mandatory)	48
Packet data field (mandatory)	variable

c. The source packet shall consist of at least 7 and at most 65542 octets.

d. A source packet which contains idle data in its packet data field is called an idle packet. Idle packets may be generated by the on-board data system when needed to maintain synchronization of the data transport and the packet extraction processes.

e. A series of source packets generated consecutively by a single application process may be designated as a group of source packets.

Figure 3-1 shows the format of the source packet as specified above including the sub-formats to be specified in the following sections.

3.1 PACKET PRIMARY HEADER

The packet primary header is mandatory and shall consist of the four fields, positioned contiguously, in the following sequence:

	Length in bits
Version number	3
Packet identification	13
Packet sequence control	16
Packet data length	16

3.1.1 <u>Version Number.</u>

a. The version number shall be contained within the bits 0–2 of the packet primary header.

b. This 3-bit field shall identify the data unit as a source packet and shall be set to "000".

The version number is used to reserve the possibility of introducing other data structures.

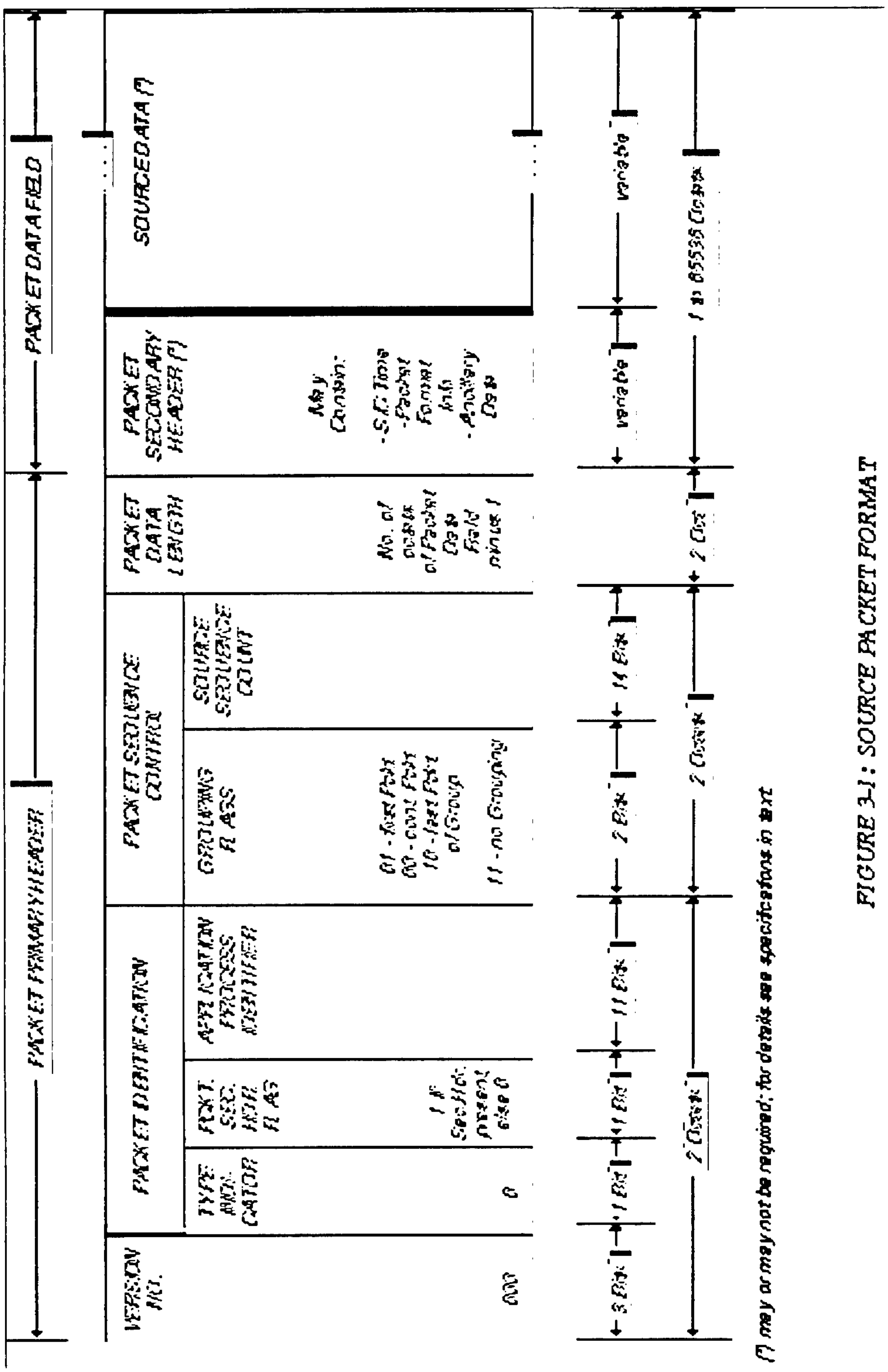

Figure 3-1. Source packet format.

3.1.2 <u>Packet Identification Field.</u>

a. The packet identification field shall be contained within the bits 3–15 of the packet primary header.

b. This 13-bit field shall be separated into three sub-fields:

	Length in bits
Type indicator	1
Packet secondary header flag	1
Application process identifier	11

The packet identification verifies the type of the packet, indicates whether the packet carries a secondary header or not, and provides information on the source of the data, i.e., the application process.

3.1.2.1 <u>Type Indicator.</u>

Bit 3 of the packet primary header shall contain the type indicator indicating the type of data unit. The type indicator shall be set to “0.”

3.1.2.2 <u>Packet Secondary Header Flag.</u>

a. Bit 4 of the packet primary header shall contain the packet secondary header flag.

b. The packet secondary header flag shall indicate the presence or absence of the packet secondary header within this source packet. It shall be “1”, if a packet secondary header is present; it shall be “0”, if a packet secondary header is not present.

c. The packet secondary header flag shall be static with respect to the application process identifier throughout a mission phase.

d. The packet secondary header flag shall be set to “0” for idle packets.

3.1.2.3 <u>Application Process Identifier.</u>

a. Bits 5–15 of the packet primary header shall contain the application process identifier.

b. The application process identifier shall be different for different application processes on the same master channel (for the definition of the master channel see item 5.d).

c. For idle packets the application process indentifier shall be “11111111111”, i.e., “all ones”.

This identifier is tailored to local mission needs and is therefore assigned by mission management. Users should note that ground data accounting considerations might limit the number of different application processes, which may be active simultaneously. Certain application process Identifiers have been reserved for specific purposes as shown on tne next page.

Application ID (Decimal Equivalent)	*Utilization*
2047	*RESERVED TO IDENTIFY A "idle" PACKET*
2046	*RESERVED BY CCSDS TO IDENTIFY A FLOW OF ENCAPSULATED ISO 8473 PACKETS*
2032-2045	*RESERVED BY CCSDS FOR POSSIBLE FUTURE USE*
0-2031	*AVAILABLE FOR USER DOMAIN ASSIGNMENT BY PROJECT ORGANIZATIONS*

3.1.3 <u>Packet Sequence Control Field.</u>

a. The packet sequence control field shall be contained within bits 16–31 of the packet primary header.

b. This 16-bit field shall be sub-divided into two sub-fields as follows:

	Length in bits
Grouping flags	2
Source sequence count	14

The packet sequence control field provides a sequential count of the packets generated with the same application process identifier, and if the grouping feature is applied, provides information on the position of a source packet in a group.

3.1.3.1 <u>Grouping Flags.</u>

a. Bits 16 and 17 of the packet primary header shall contain the grouping flags.

b. The grouping flags shall be set as follows:

"01" for the first source packet of a group;
"00" for a continuing source packet of a group;
"10" for a last source packet of a group.

c. For a source packet not belonging to a group of source packets the grouping flags shall be set to "11".

d All source packets belonging to a specific group of source packets shall originate from the same application process identified by a unique application process identifier.

The use of a group of source packets is outside the scope of this recommendation.

3.1.3.2 Source Sequence Count.

a. Bits 18–31 of the packet primary header shall contain the source sequence count.
b. The source sequence count shall provide the sequential binary count of each source packet generated by an application process identified by a unique application process identifier.
c. The source sequence count shall be continuous, modulo 16384.
d. Idle packets are not required to increment the source sequence count.
e. A re-setting of the source sequence count before reaching 16383 shall not take place unless it is unavoidable.

The purpose of the field is to order this packet with other packets generated by the same application process, even though their natural order may have been disturbed during transport to the on-board recording device and subsequent playback to the user's processor on the ground.

The field will normally be used in conjunction with a time code (see paragraph 3.2.1.1) to provide unambiguous ordering; it is therefore essential that the resolution of the time code is sufficient for this code to increment at least once between successive recycling of the source sequence count.

If the source sequence count is re-set due to an unavoidable re-initialization of a process the completeness of a sequence of source packets cannot be determined.

3.1.4 Packet Data Length Field.

a. The packet data length field shall be contained within bits 32–47 of the packet primary header.
b. This 16-bit field shall contain a binary number equal to the number of octets in the packet data field minus 1.
c. The value contained in the packet data length field may be variable and shall be in the range of 1 to 65536 octets.

Users should recognize that although very long packets are permissible, these may present special problems in terms of recording system monopolization and source data buffering and may add complexity to ground processing. The standard therefore provides the means to assign these packets to individual virtual channels (see Chapter 5). An additional measure could be to limit the maximum length of the source packets for a specific mission or mission phase.

3.2 Packet Data Field

a. The packet data field shall follow, without gap, the packet primary header.
b. The packet data field is mandatory and shall consist of at least one of the two fields, positioned contiguously, in the following sequence:

	Length in octets
Packet secondary header	variable
Source data field	variable

c. The packet data field shall contain at least one octet.

3.2.1 Packet Secondary Header.

a. If present, the packet secondary header shall follow, without gap, the packet data length field.
b. The packet secondary header is mandatory if no source data field is present; otherwise it is optional. The presence or absence of a packet secondary header shall be signaled by the packet secondary header flag within the packet identification field (see paragraph 3.1.2.2).
c. If present, the packet secondary header field shall consist of:

Packet secondary header identification field,*
Packet secondary header time code preamble field,**
Packet secondary header time code field (optional), and
Packet secondary header data field (optional).

*Mandatory
**Mandatory if time code field is present.

The chosen options shall remain static for a specific application process identifier throughout all mission phases.

a. If present, the packet secondary header shall contain a packet secondary header identification field. The packet secondary header identification field shall be separated into two sub-fields:

	Length in bits	location (bit #)
Time code flag	1	0
Packet secondary header field length	7	1-7

b. The time code flag shall be a value of "1" to indicate a time code format is present.
c. The packet secondary header field length indicates the number of octets in the packet secondary header minus 1.

1

The purpose of the secondary header is to allow (but not require) a means for placing ancillary data (time, internal data field format.) within a source packet. The maximum length of the packet secondary header shall be 127 octets.

3.2.1.1 **Packet Secondary Header Time Field.**

a. The "time code format" shall consist of a preamble field (P-field) and a time field (T-field). The time code format shall consist of an integral number of octets.
b. The preamble field shall consist of one of the CCSDS P-fields for segmented binary time codes specified in appendix C. The P-field extension flag (bit-0) shall be set to "0" indicating that there is not a second P-field octet present.
c. The packet secondary header time code field shall consist of one of the CCSDS segmented binary time codes specified in appendix C.

The time codes defined in appendix C consist of a P-field, which identifies the time code and its characteristics, and a T-field.

d. The time code selected shall be static for a given application process identifier throughout all mission phases.

For services such as archiving, sorting, processing and correlation with other data sets, the source sequence count may have to be concatenated with a time field in order to identify a packet unambiguously.

See also the comment concerning time code under paragraph 3.1.3.2.

3.2.1.2 **Packet Secondary Header Data Field.**

The packet secondary header data field shall consist of an integral number of octets.

The data field may contain any ancillary data necessary for the interpretation of the information contained within the source data field of the packet. The content and the format of this data are not specified by this standard.

3.2.2 **Source Data Field.**

a. If present, the source data field shall follow, without gap, either the packet secondary header (if a packet secondary header is present) or the packet data length field (if a packet secondary header is not present).
b. The source data field is mandatory if no packet secondary header is present, otherwise it is optional.
c. The source data field shall contain either source data from an application process or idle data.
d. The length of the source data field may be variable. It shall contain an integral number of octets. See also the specifications in sub-section 3.1.4.

4. [NO LONGER USED]

[This chapter defined the source packet segment, which is no longer defined in this issue.]

5. TRANSFER FRAME

a. The transfer frame shall provide the data structure for the transmission of (1) source packets, and (2) idle data across the data recording channel.

b. The transfer frame shall encompass the major fields, positioned contiguously, in the following sequence:

	Length in bits
Transfer frame primary header (mandatory)	48
Transfer frame secondary header (optional)	16, 24, ... or 512
Transfer frame data field (mandatory)	variable
Operational control field (not used)	32
Frame error control field (mandatory)	16

c. The transfer frame shall be of constant length throughout a specific mission phase.

The maximum length of a transfer frame data field is limited by reference [3] to 8,920 bits. See appendix B for excerpts of the appropriate sections of reference [3] being adopted by this standard.

d. All transfer frames with the same transfer frame version number (see sub-section 5.1.1) and the same vehicle identifier (see paragraph 5.1.2.1) on the same physical channel constitute a master channel.

In most cases the master channel will be identical with the physical channel. However, when the physical channel also carries transfer frames with other vehicle identifiers a distinction between master channel and physical channel is necessary, i.e., multiplexing of transfer frames with different vehicle identifiers will be performed by the multiplexing of different master channels on the same physical channel.

e. A master channel shall consist of between one and eight virtual channels.

Although packet telemetry systems may be designed to tolerate channel noise, full benefit from packet telemetry will require that a high-quality data channel be provided so that packetized data may be adaptively inserted into the frame. The synchronization bits (Attached Sync Marker (ASM)) shall adhere to the requirements of reference [3]. See appendix B for excerpts of the appropriate sections of reference [3] being adopted by this standard.

Figure 5-1 illustrates the detailed format of the transfer frame.

5.1 TRANSFER FRAME PRIMARY HEADER

The transfer frame primary header is mandatory and shall consist of five fields, positioned contiguously, in the following sequence:

	Length in bits
Transfer frame version number	2
Transfer frame identification	14
Master channel frame count	8
Virtual channel frame count	8
Transfer frame data field status	16

The primary header covers five principal functions:

Identification of the data unit as a transfer frame,
Identification of the vehicle (and possibly of the link, if applicable) that recorded the data,
Multiplexing of the virtual channels into one master channel,
Providing a counting mechanism for the virtual channels and the master channel, and
Providing pointers and other control information so that a variable length source packet may be extracted from the transfer frame data field.

5.1.1 Transfer Frame Version Number.

a. The transfer frame version number shall be contained within bits 0–1 of the transfer frame primary header.
b. This 2-bit field shall identify the data unit as a transfer frame; it shall be set to "00".

This standard defines Version 1 of the transfer frame.

5.1.2 Transfer Frame Identification Field.

a. The transfer frame identification field shall be contained within bits 2–15 of the transfer frame primary header.
b. This 14-bit field shall be sub-divided into three sub-fields as follows:

	Length in bits
Vehicle identifier	10
Virtual channel identifier	3
Operational control field flag	1

This field identifies the generator of the transfer frame, it specifies the virtual channel to which it belongs, and it provides information on the format of the transfer frame.

5.1.2.1 Vehicle Identifier.

a. Bits 2–11 of the transfer frame primary header shall contain the vehicle identifier.
b. The vehicle identifier shall provide the identification of the vehicle that created the frame of data.
c. The vehicle identifier shall be static throughout all mission phases.

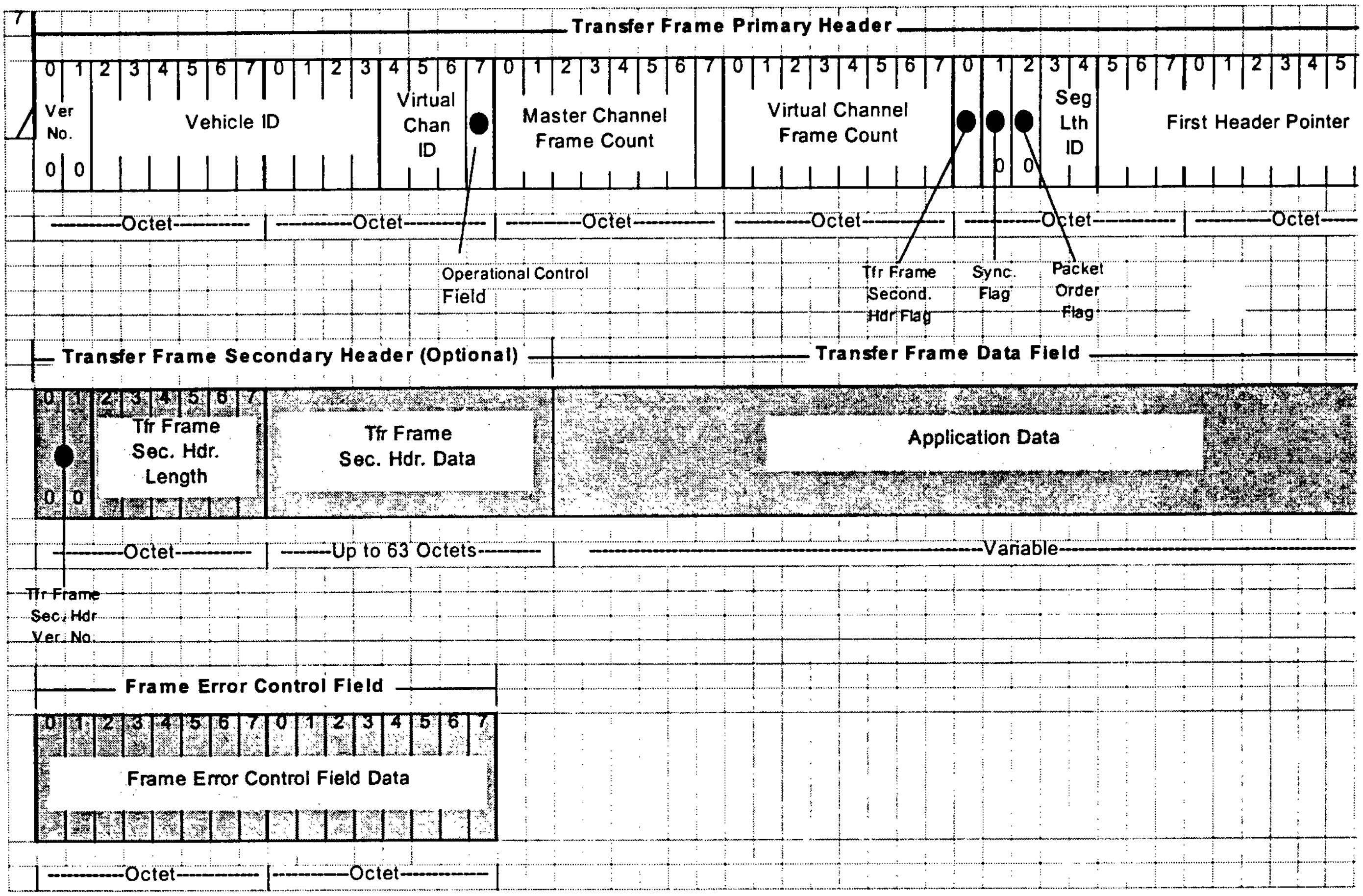

Figure 5-1. Transfer frame format.

5.1.2.2 Virtual Channel Identifier.

a. Bits 12–14 of the transfer frame primary header shall contain the virtual channel identifier.
b. The virtual channel identifier provides the identification of the virtual channel.

The order of occurrence of different virtual channels on a master channel may vary.

5.1.2.3 Operational Control Field Flag.

Bit 15 of the transfer frame primary header shall contain the operational control field flag.

(For purposes of this standard, the operational control field will not be used and the operational control field flag shall be set to "0" (operational control field, not present)).

5.1.3 Master Channel Frame Count Field.

a. The master channel frame count field shall be contained within bits 16–23 of the transfer frame primary header.
b. This 8-bit field shall contain a sequential binary count (modulo 256) of each transfer frame transmitted within a specific master channel.
c. A re-setting of the master channel frame count before reaching 255 shall not take place unless it is unavoidable.

The purpose of this field is to provide a running count of the frames that have been transmitted through the same master channel.

If the master channel frame count is re-set due to an unavoidable re-initialization, the completeness of a sequence of transfer frames cannot be determined.

5.1.4 Virtual Channel Frame Count Field.

a. The virtual channel frame count field shall be contained within bits 24–31 of the transfer frame primary header.
b. This 8-bit field shall contain a sequential binary count (modulo 256) of each transfer frame transmitted through a specific virtual channel of a master channel.
c. A re-setting of the virtual channel frame count before reaching 255 shall not take place unless it is unavoidable.

The purpose of this field is to provide individual accountability for each of the maximum eight virtual channels, primarily to enable systematic source packet extraction from the transfer frame data field.

If the virtual channel frame count is re-set due to an unavoidable re-initialization the completeness of a sequence of transfer frames in the related virtual channel can not be determined.

5.1.5 Transfer Frame Data Field Status Field.

The transfer frame data field status field shall be contained within bits 32–47 of the transfer frame primary header.

This 16-bit field shall be sub-divided into five sub-fields as follows:

	Length in bits
Transfer frame secondary header flag	1
Synchronization flag	1
Packet order flag	1
Segment length identifier	2
First header pointer	11

This field indicates whether a secondary header is present. Further, it provides information on the type of data contained in the frame and provides, together with the virtual channel frame count, the control information necessary to enable source packets to be extracted from the transfer frame data field.

5.1.5.1 <u>Transfer Frame Secondary Header Flag.</u>

a. Bit 32 of the transfer frame primary header shall contain the transfer frame secondary header flag.
b. The transfer frame secondary header flag shall signal the presence or absence of the transfer frame secondary header. It shall be "1", if a transfer frame secondary header is present; it shall be "0", if a transfer frame secondary header is not present.
c. The transfer frame secondary header flag shall be static within a specific master channel throughout a mission phase when the transfer frame secondary header is associated with a master channel.
d. The transfer frame secondary header flag shall be static within a specific virtual channel throughout a mission phase when the transfer frame secondary header is associated with a virtual channel.

For the significance of the above mentioned associations see item 5.2.d.

5.1.5.2 <u>Synchronization Flag.</u>

a. Bit 33 of the transfer frame primary header shall contain the synchronization flag.
b. The synchronization flag shall signal the type of data, which is inserted into the transfer frame data field. It shall be "0" if octet-synchronized and forward-ordered source packets or idle data are inserted; it shall be "1" if privately defined data is inserted. For purposes of this standard privately defined data shall not be used and the synchronization flag shall always be set to "0."
c. The synchronization flag shall be static within a specific virtual channel throughout a mission phase.

Source packet data units are normally inserted into the transfer frame data field synchronously on octet boundaries one following directly after another. Generally, the source packets "spill over" into the next frame for the same virtual channel; therefore, source packets do not usually begin at the first octet of the transfer frame data field. The location of the first source packet

header in a particular transfer frame is identified by the first header pointer field. (See comment under paragraph 5.1.5.5 also.)

5.1.5.3 Packet Order Flag.

a. Bit 34 of the transfer frame primary header shall contain the packet order flag.
b. The packet order flag is reserved for future use and shall be set to "0."

5.1.5.4 Segment Length Identifier.

a. Bits 35 and 36 of the transfer frame primary header shall contain the segment length identifier.
b. The segment length identifier shall be set to "11."

5.1.5.5 First Header Pointer.

a. Bits 37–47 of the transfer frame primary header shall contain the first header pointer.
b. If the synchronization flag is set to "0," the first header pointer shall contain information on the position of the first source packet within the transfer frame data field.
c. The locations of the octets in the transfer frame data field shall be numbered in ascending order. The first octet in this field is assigned the number 0. The first header pointer shall contain the binary representation of the location of the first octet of the first packet primary header.

The locations of any subsequent headers within the same transfer frame data field will be determined by calculating these locations using the packet data length field. (See subsection 3.1.4.)

The specification also covers the following two special cases:

(1) If a first source packet primary header starts at the end of the transfer frame data field within frame N and spills over into frame M of the same virtual channel, the first header pointer in frame N indicates the start of this header.

(2) If a source packet header is split between frames N and M (M>N), the first header pointer in frame M ignores the residue of the split header and only indicates the start of any subsequent new source packet header within frame M.

In both cases (1) and (2) above, one or more frames with idle data may occur between frame N and frame M (M>N).

d. If no packet primary header starts in the transfer frame data field, the first header pointer shall be set to "11111111111."
e. If a transfer frame contains idle data in its transfer frame data field, the first header pointer shall be set to "11111111110."

5.2 TRANSFER FRAME SECONDARY HEADER

a. If present, the transfer frame secondary header shall follow, without gap, the transfer frame primary header.
b. The transfer frame secondary header is optional; its presence or absence shall be signaled by the transfer frame secondary header flag in the transfer frame primary header. (See paragraph 5.1.5.1.)
c. The transfer frame secondary header shall consist of an integral number of octets as follows:

	Length in bits
Transfer frame secondary header identification field	8
Transfer frame secondary header data field	8, 16, or 504

d. The transfer frame secondary header shall be associated with either a master channel or a virtual channel.

The association of a secondary header with a master channel allows data to be transferred frame-synchronously with respect to this master channel.

e. The transfer frame secondary header shall be of fixed length within the associated master channel or within the associated virtual channel throughout a mission phase.

5.2.1 <u>Transfer Frame Secondary Header Identification Field.</u>

a. The transfer frame secondary header identification field shall be contained within bits 0–7 of the transfer frame secondary header.
b. The transfer frame secondary header identification field shall be sub-divided into two sub-fields as follows:

	Length in bits
Transfer frame secondary header version number field	2
Transfer frame secondary header length field	6

5.2.1.1 <u>Transfer Frame Secondary Header Version Number.</u>

a. The transfer frame secondary header version number shall be contained within bits 0–1 of the transfer frame secondary header.
b. The transfer frame secondary header version number shall be set to "00."

This sub-field shall indicate which of up to four secondary header versions is used. The present standard recognizes only one version.

5.2.1.2 <u>Transfer Frame Secondary Header Length.</u>

a. The transfer frame secondary header length shall be contained within bits 2–7 of the transfer frame secondary header.
b. This sub-field shall contain the total length of the transfer frame secondary header in octets minus one, represented as a binary number.
c. The transfer frame secondary header length shall be static either within a specific master channel or a specific virtual channel throughout a mission phase.

When a secondary header is present, this length may be used to compute the location of the start of the frame data field.

5.2.2 <u>Transfer Frame Secondary Header Data Field.</u>

a. The transfer frame secondary header data field shall follow, without gap, the transfer frame secondary header identification field.
b. The transfer frame secondary header data field shall contain the transfer frame secondary header data.

The field could be used for a time code (see paragraph 3.2.1.1; its insertion is, however, not mandatory) to facilitate seeking start/stop times during ground playback. See appendix A for possible uses for this field.

5.3 TRANSFER FRAME DATA FIELD

a. The transfer frame data field shall follow, without gap, the transfer frame primary header or the transfer frame secondary header if present.
b. The transfer frame data field shall contain the data to be recorded on-board the vehicle and shall consist of an integral number of octets. Transfer frame data may be any of the three types of data specified in item 5.a.
c. Source packets shall be inserted contiguously and in forward order into the transfer frame data field.
d. The length of the transfer frame data field shall be constrained by the length of the total transfer frame. For this constraint see item 5.c.
e. Source packets may either be transmitted on separate virtual channels or may be mixed on the same virtual channel.
f. In the case where not sufficient source packets (including idle packets) are available to fill a transfer frame data field, a transfer frame with a data field containing only idle data shall be transmitted.

Transfer frames containing idle data in their data fields may have to be sent to maintain synchronization with the recording device and also because the secondary header may still be needed to transmit valid data.

Transfer frames carrying idle data may be sent on a virtual channel that also carries packets, but a separate virtual channel dedicated to idle data is preferred.

Idle data in a transfer frame data field must not be confused with idle packets specified in item 3.d.

Packets with different application process identifiers may be multiplexed in the frame data field in any combination.

5.4 OPERATIONAL CONTROL FIELD

(The operational control field is not supported by this issue of the standard.)

5.5 FRAME ERROR CONTROL FIELD

a. The frame error control field shall occupy the two octets following, without gap, the transfer frame data field.
b. The frame error control field is mandatory.
c. The frame error control field shall occur within every transfer frame transmitted within the same master channel throughout a mission phase. If the frame error control field is not utilized for error detection purposes, it is recommended to fill the field with all ones or zeros.

The purpose of this field is to provide a capability for detecting errors that have been introduced into the frame during the data handling process.

5.5.1 Encoding Procedure.

a. The encoding procedure accepts an (n-16)-bit transfer frame, excluding the frame error control field, and generates a systematic binary (n, n-16) block code by appending a 16-bit frame error control field as the final 16 bits of the codeblock.
b. The equation for the contents of the frame error control field is:

$$\text{FECF} = [(X^{16} \cdot M(X)) + (X^{(n-16)} \cdot L(X))] \text{ modulo } G(X)$$

where
— all arithmetic is modulo 2;
— n is the number of bits in the encoded message;
— M(X) is the (n-16)-bit message to be encoded expressed as a polynomial with binary coefficients;
— L(X) is the presetting polynomial given by

$$L(X) = \sum_{i=0}^{15} X^i \ ;$$

— G(X) is the generating polynomial given by:

$$G(X) = X^{16} + X^{12} + X^{5} + 1.$$

The $X^{(n-16)} \cdot L(X)$ term has the effect of presetting the shift register to all "1" state prior to encoding.

5.5.2 Decoding Procedure.

The error detection syndrome, S(X), is given by

$$S(X) = [(X^{16} \cdot C^{*}(X)) + (X^{n} \cdot L(X))] \text{ modulo } G(X)$$

where
$C^{*}(X)$ is the received block, including the frame error control field, in polynomial form
S(X) is the syndrome polynomial which will be zero if no error is detected and non-zero if an error is detected.

INDEX

APPENDIX A

DIGITAL DATA ACQUISITION AND ON-BOARD RECORDING: SUMMARY OF CONCEPT AND RATIONALE

This appendix presents the conceptual framework and rationale for the RCC Digital Data Acquisition and On-Board Recording System. The background information provided here will be found helpful in understanding the RCC Digital Data Acquisition and On-Board Recording Standard. Through the process of normal evolution, it is expected that expansion, deletion or modification to this report may occur. Questions relative to the contents or status of this report should be addressed to the RCC Secretariat.

TABLE OF CONTENTS

Figures

APPENDIX A

1. PURPOSE, SCOPE AND ORGANIZATION

This report contains the concept and supporting rationale for the Digital Data Acquisition and On-Board Recording System developed by the Telemetry Group (TG) of the Range Commanders Council (RCC). It has been prepared to provide an introduction and overview of the system concept upon which the detailed RCC Digital Data Acquisition and On-Board Recording Standard is based. The Digital Data Acquisition and On-Board Recording Standard is based on the Consultative Committee for Space Data Systems (CCSDS) Packet Telemetry Recommendation (reference [1]). (The concept and supporting rationale for the CCSDS Packet Telemetry Recommendation is contained in reference [4]).

Currently there are a large number of unique data structures that have been developed by vendors and government for specific on-board data recording applications. These unique data structures require unique decoding software programs. Writing unique decoding software, checking the software for accuracy, and decoding the data tapes is extremely time consuming and costly.

A need was identified to develop a digital data acquisition and on-board recording standard that will support the multiplexing of multiple data streams and maintain the accuracy of data correlation with time. Specifically, the Digital Data Acquisition and On-Board Recording Standard should be compatible with the multiplexing of both synchronous and asynchronous digital inputs such as Pulse Code Modulation (PCM), MIL-STD-1553 asynchronous data bus, digital voice, time, discrete, video, and RS-232/422 communication data. In addition, the Digital Data Acquisition and On-Board Recording Standard should be aligned with current developments in layered communications architecture.

The digital data acquisition standard should allow the use of a common set of playback/data reduction software, take advantage of emerging random access recording media, improve efficiency over traditional PCM for asynchronous data, and take advantage of the rapid improvement in commercial communication technology.

1.2 SCOPE

The concepts, protocols and data formats developed for the CCSDS Telemetry System and adopted for the RCC Digital Data Acquisition and On-Board Recording System described herein are designed for flight and ground data systems supporting conventional, contemporary, instrumented vehicles. Data formats are designed with efficiency as a primary consideration, i.e., format overhead is minimized. (A report on the analysis of candidate data structures as a possible Digital Data Acquisition and On-Board Recording Standard is contained in reference [5]).

The CCSDS Packet Telemetry Recommendation is primarily intended to support multi-point to multi-point data transfer over space based transmission links. In contrast, the RCC Digital Data Acquisition and On-Board Recording Standard is intended to support point-to-point data acquisition and recording and subsequent play back on a ground-based data acquisition system. While there are significant similarities, the differences require some minor deviations from the absolute adoption of the CCSDS Packet Telemetry Recommendation. It is therefore the intent of the RCC Digital Data Acquisition and On-Board Recording Standard to duplicate the CCSDS Packet Telemetry Recommendation and deviate only when unique on-board recording/ground playback or point-to-point considerations warrant.

1.3 ORGANIZATION

An overview of the RCC Digital Data Acquisition and On-Board Recording System is presented in section 2. This discussion introduces the notion of architectural layering to achieve transparent and reliable delivery of scientific and engineering sensor data (generated aboard remote vehicles) to an on-board recording device and subsequently played back through ground data systems to a user on the ground. Section 3 presents a more detailed description of the Digital Data Acquisition and On-Board Recording System and rationale for the specific RCC Digital Data Acquisition and On-Board Recording Standard.

Annex A-1 provides a glossary to familiarize the reader with the terminology used throughout the RCC Digital Data Acquisition and On-Board Recording System. Annex A-2 contains application notes that describe how a project may implement complete or partial compatibility with the RCC Digital Data Acquisition and On-Board Recording Standard. Annex A-3 is a guideline for transfer frame error detection coding.

2. OVERVIEW OF THE RCC DIGITAL DATA ACQUISITION AND ON-BOARD RECORDING SYSTEM

2.1 INTRODUCTION

The purpose of a Digital Data Acquisition and On-Board Recording System is to reliably and transparently convey measurement information from a data-generating source to a recording device and then to playback the data to user(s) located on the ground. Typically, data generators are scientific sensors, science housekeeping sensors, engineering sensors and other subsystems on-board a vehicle.

The advent of capable microprocessor based hardware will result in data systems with demands for greater throughput and a requirement for corresponding increases in vehicle autonomy and mission complexity. These facts, along with the current technical and fiscal environments, create a need for greater Digital Data Acquisition and On-Board Recording capability and efficiency with reduced costs.

The lack of effective standardization among various missions and ranges forces "multi-mission/multi-range" data acquisition systems to require significant set-up costs/time to handle the variety of unique on-board data recording formats. Higher level data handling services oriented more toward computer-to-computer transfers and typical of modern day commercial and military communication networks must be custom designed and implemented on a mission by mission basis. The intent of the Digital Data Acquisition and On-Board Recording System is not only to ease the transition toward greater automation within organizations, but also to promote standardization among the organizations, thereby resulting in greater cross-support opportunities and services.

Digital Data Acquisition and On-Board Recording: packet telemetry is a concept which facilitates the on-board recording of remote vehicle-acquired data from source to on-board recorder and then playback to the user via a ground data acquisition system in a standardized and highly automated manner. Packet telemetry provides a mechanism for implementing common data structures and protocols that can enhance the development and operation of range missions. Packet telemetry for Digital Data Acquisition and On-Board Recording addresses the process of the end-to-end transport of DOD mission data sets from source application processes located on a vehicle to an on-board recording device, and then played back from the recording device to user application processes located on the ground.

The Digital Data Acquisition and On-Board Recording Standard is primarily concerned with describing the on-board recording formats which are generated by remote vehicles in order to execute their roles in the above processes.

Packet telemetry services provide the user with reliable and transparent recording and playback of remote vehicle digital data.

2.2 DIGITAL DATA ACQUISITION AND ON-BOARD RECORDING SYSTEM CONCEPT

The system design technique known as layering was found to be a very useful tool for transforming the Digital Data Acquisition and On-Board Recording System concept into sets of operational and formatting procedures. The layering approach is patterned after the International Organization for Standardization's Open Systems Interconnection Reference Model for networking (reference [6]), which is a seven layered architecture that groups functions logically and provides conventions for passing information from layer to layer. Layering decomposes a complex procedure into sets of peer functions residing in common architectural strata with standardized interfaces between them.

Within each layer, the functions exchange data according to established standard rules or "protocols." Each layer draws upon a well-defined set of services provided by the layer below, and provides a similarly well defined set of services to the layer above. As long as these service interfaces are preserved, the internal operations within a layer are unconstrained and transparent to other layers. Therefore, an entire layer within a system may be removed and replaced as dictated by user or technological requirements without destroying the integrity of the rest of the system. Further, as long as the appropriate interface protocol is satisfied, a customer (user) can interact with the system/service at any of the component layers. Layering is therefore a powerful tool for designing structured systems that change due to the evolution of requirements or technology.

A companion standardization technique that is conceptually simple, yet very robust, is the encapsulation of data within an envelope or "header." The header contains the identifying information needed by the layer to provide its service while maintaining the integrity of the envelope contents.

Figure 2-1 illustrates the RCC Digital Data Acquisition and On-Board Recording System in terms of a layered service model. It should be noted that the RCC Digital Data Acquisition and On-Board Recording Standard only addresses two of the layers (packetization and transfer) of this model. ("Application Notes" presented in Annex B detail implementation guidelines for compatibility with the layered services at the interface between application data and the packetization layer and between transfer frames and the coding layer).

Figure 2-1 is modified from the CCSDS Packet Telemetry Recommendation by showing a recording device as the "recipient" of the physical layer waveform. It also depicts applicability to only the packetization and transfer layers.

2.2.1 Packetization Layer.

Within the Digital Data Acquisition and On-Board Recording Standard, packet telemetry vehicle-generated application data are formatted into end-to-end transportable data units called source packets. These data are encapsulated within a primary header that contains identification, sequence control and packet length information. A source packet is the basic data unit recorded

on-board the vehicle and generally contains a meaningful quantity of related measurements from a particular source.

2.2.2. Segmentation Layer.

[NO LONGER USED - PER ISSUE 4, CCSDS PACKET TELEMETRY RECOMMENDATION]

2.2.3 Transfer Frame Layer. The transfer frame is used to reliably multiplex source packets onto the on-board recording device. As the heart of the RCC Digital Data Acquisition and On-Board Recording System, the transfer frame protocols offer a range of delivery service options. An example of such a service option is the multiplexing of transfer frames into virtual channels (VCs).

The transfer frame begins with an attached frame synchronization marker and is followed by a primary header. The primary header contains frame identification, channel frame count information, and frame data field status information.

The transfer frame data field is followed by a frame error control field. The frame error control field provides the capability for detecting errors that may have been introduced into the frame during the data handling process. The delivery of transfer frames requires the services provided by the lower layers (e.g., coding/decoding) to accomplish its role.

2.2.4 Channel Coding Layer.

The channel coding layer is not specified in the RCC Digital Data Acquisition and On-Board Recording Standard although a CCSDS Channel Coding Recommendation, reference [2], is included in the CCSDS Packet Telemetry Recommendation. Data errors and/or loss on recording are significantly less than experienced in over-the-air transmission of telemetry signals. Therefore, the specific coding implementation is left to the discretion of the implement depending on the characteristics of the physical recording medium. It is recommended that channel coding for purposes of error detection and/or correction be integrated into the recording device and be transparently removed during playback. Appendix B contains those portions of the CCSDS Channel Coding Recommendation that are applicable to the Digital Data Acquisition and On-Board Recording Standard. (Future versions of the RCC Digital Data Acquisition and On-Board Recording Standard may specify a coding layer standard).

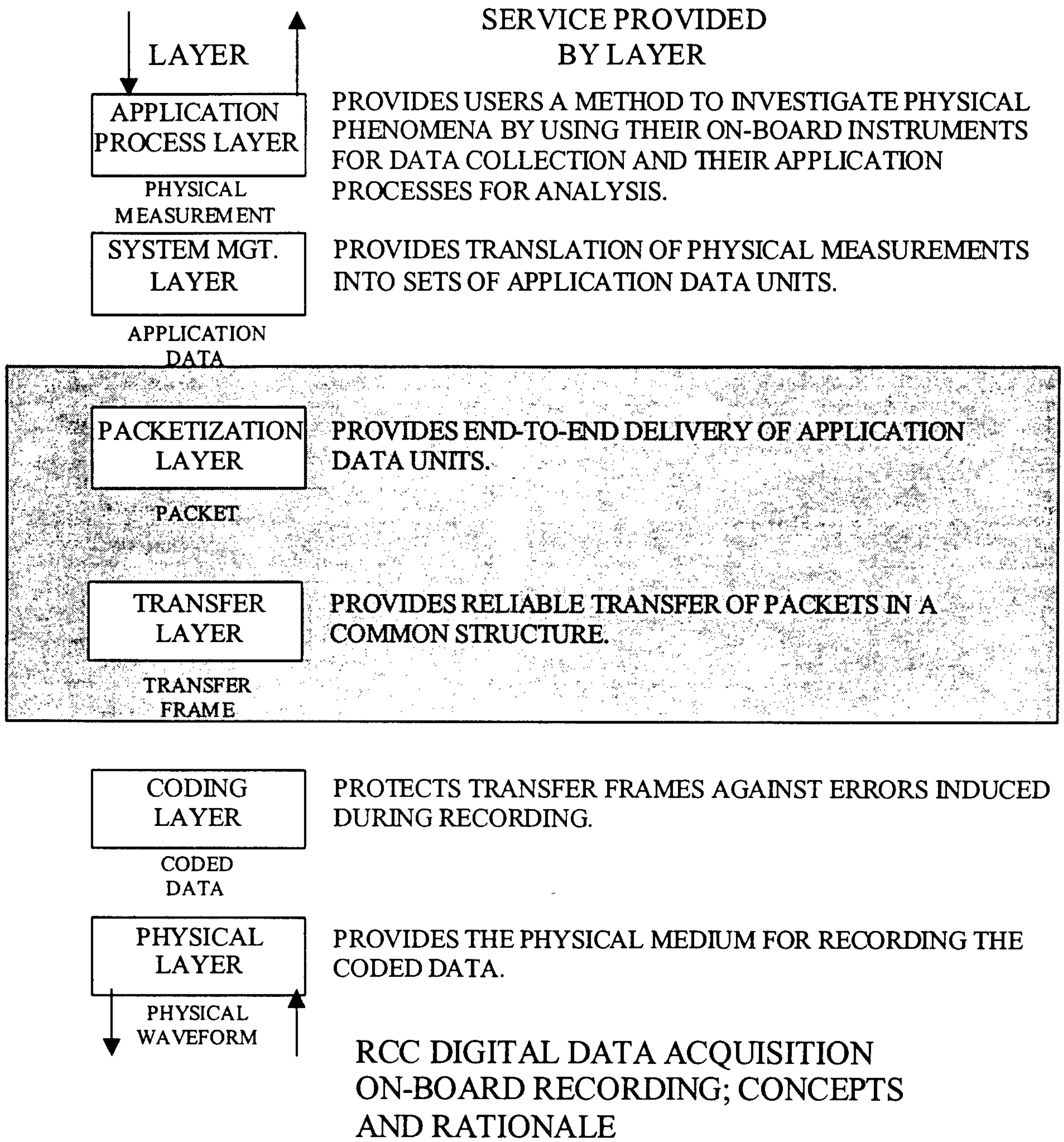

Figure 2-1: Layered digital data acquisition and on-board recording service model.

2.2.5 Relationship Between Telemetry and Telecommand Systems.

The RCC Digital Data Acquisition and On-Board Recording Standard is for one-way communication from the vehicle to the on-board recording device and from the playback device to the ground data acquisition system. There are no provisions for telecommands (uplink).

3. DIGITAL DATA ACQUISITION AND ON-BOARD RECORDING SYSTEM DESCRIPTION AND RATIONALE

This section describes the services and protocols characterizing the Digital Data Acquisition and On-Board Recording System and presents the rationale for detailed structure of the data units. The section discusses the two main protocol and format areas: 1) source packet and 2) transfer frame.

3.1 PACKET RECORDING

3.1.1 <u>Introduction.</u> Packet telemetry represents an evolutionary step from the traditional Time-Division Multiplex (TDM) method of acquiring, recording, and playing back scientific applications and engineering data from instrumented vehicle sources to ground data systems sinks. It also relies on a layered architectural model to isolate independent interfaces. The packet telemetry process has the conceptual attributes of

(1) Facilitating the acquisition, transmission, and recording of instrument data at a rate appropriate for the phenomenon being observed.
(2) Defining a logical interface and protocol between an instrument and its associated on-board recorder/playback and ground support equipment which remains constant throughout the life cycle of the instrument (bench test, integration, flight, and possible re-use).
(3) Simplifying overall system design by allowing microprocessor-based symmetric design of the instrument control and data paths ("transfer frames in," transfer frames out") compatible with commercially available components and interconnection protocol standards.
(4) Facilitating interoperability of instrumented vehicle systems whose data acquisition and on-board recording systems interfaces conform to CCSDS/IRIG guidelines.
(5) Enabling the delivery of high-quality data products in a mode that is faster and less expensive than would be possible with conventional methods.

Figure 3-1 is a functional diagram of the Digital Data Acquisition and On-Board Recording System data flow from the creation of a data set by an application process operating within a vehicle "source" (instrument or subsystem), through to the delivery of the same data to a user "sink" (application process) on the ground. Since many of the elements of this flow are presently mission-unique, a primary objective of packet telemetry is to define stable, mission-independent interface standards for the communications path within the flow.

Figure 3-1 is modified from the CCSDS Packet Telemetry Recommendation by showing the transfer frames recorded onto a physical medium and then played back from the physical medium, instead of the over-the-air transmission of the coded transfer frames.

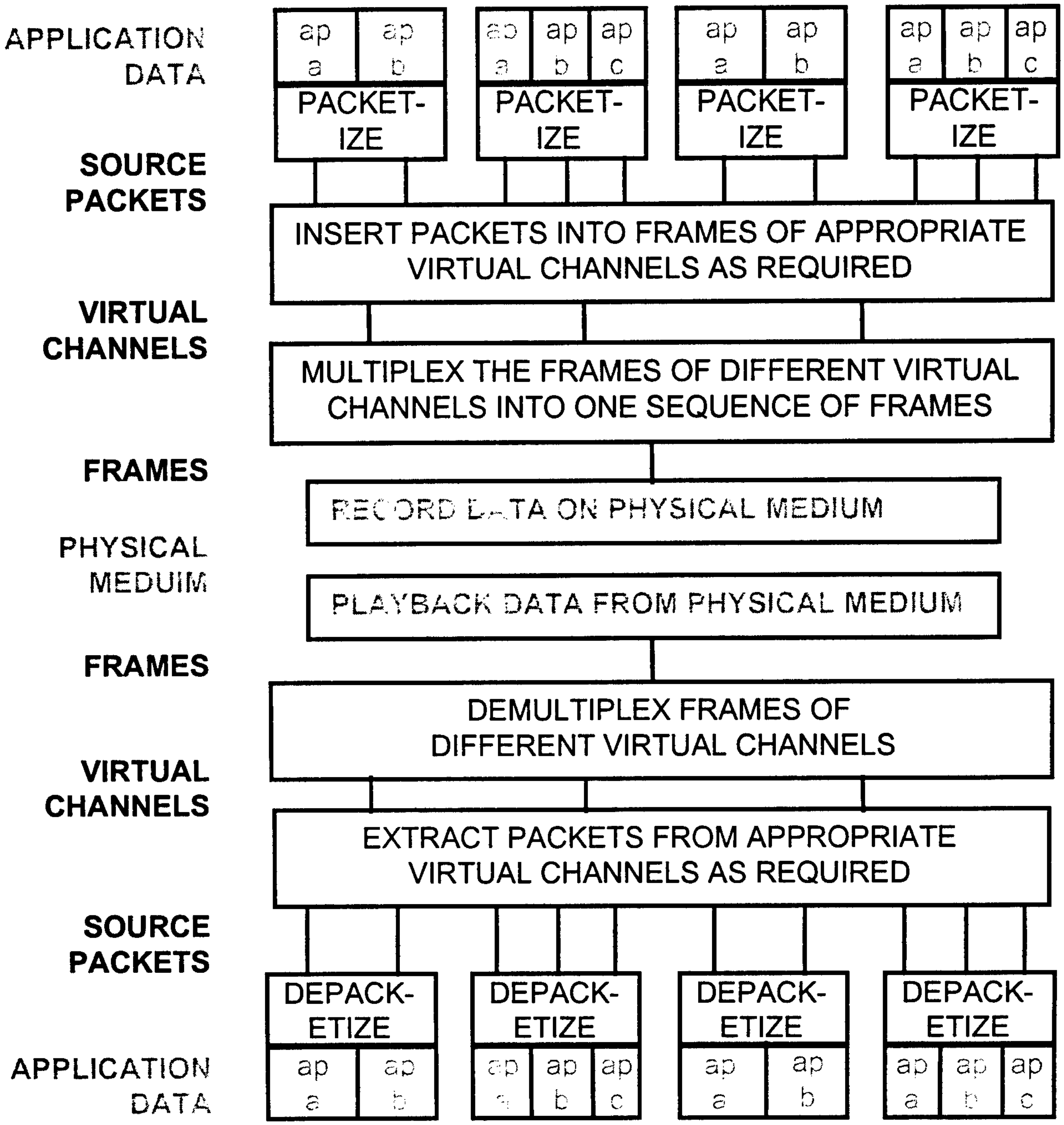

Figure 3-1. Digital data acquisition and on-board recording system data flow.

3.1.2 SOURCE PACKET. A source packet is a data unit encapsulating a block of observational data that may include ancillary data and may be directly interpreted by the receiving end application process. The source packet format (version 1), with the addition of a secondary header and packet error control field, is reproduced in Figure 3-2 below for the convenience of the reader.

The source packet data structure of Figure 3-2 is identical to the source packet data structure of the CCSDS Packet Telemetry Recommendation (reference [1]).

User application data are encapsulated within a packet by prefacing them with a standard label or "primary header," which is used by the data transport system to route the data through the system and to allow the user to reconstruct the original data set. The primary header consists of three main fields: packet identification, packet sequence control and packet length.

PRIMARY HEADER							DATA FIELD	
PACKET IDENTIFICATION				PACKET SEQUENCE CONTROL		PACKET LENGTH	SECONDARY HEADER (OPTIONAL)	SOURCE DATA
VERSION #	TYPE	SEC. HDR. FLAG	APPLIC. PROCESS ID	GROUPING FLAGS	SOURCE SEQUENCE COUNT		ID FIELD TIME AUX DATA	
3	1	1	11	2	14			
16 BITS				16 BITS		16 BITS	(VARIABLE) UP TO 127 OCTETS)	(VARIABLE)

Figure 3-2: Source packet (version 1) format.

3.1.2.1. <u>Packet Identification.</u>

Version Number. The version number is the first of four sub-fields of packet identification. This sub-field explicitly indicates the version of the formatted packet, and its length of three bits allows eight different versions to be identified. While only two versions are currently defined, this arrangement allows a reasonable growth capability to support future needs. However, in the interest of constraining the proliferation of standards, additional versions will be discouraged unless it can be demonstrated that the current versions are truly inadequate.

Type. The second sub-field is a one-bit identifier to signal that this packet is a "Telemetry" packet and not a "telecommand" packet. It is always set to "zero" for telemetry packets.[1]

[1]In the first issue of reference [1] (May 1984) this field was described as a "reserved spare" and was, by convention, set to zero for telemetry. In issue 2 (January 1987), the value of the field has not changed, but its function has been established.

As there is no "telecommand" packet identified in the RCC Digital Data Acquisition and On-Board Recording Standard, the type is always set to "zero."

Secondary Header Flag. The third sub-field is a one-bit secondary header flag. The RCC recognizes that users may need a means of encapsulating ancillary data (such as time, internal data field format, vehicle position/attitude, etc.) which may be necessary for the interpretation of the information contained within the packet. Therefore, this flag, when set to one, indicates that a secondary header follows the primary header.

Application Process ID. The last sub-field in the packet identification field is used to uniquely identify the originating source packet application process. Eleven bits are allocated to the Application Process ID, permitting identification of up to 2048 separate application processes per vehicle, sufficient for any envisioned mission. For positive identification, one can consider this sub-field an extension of the vehicle ID, which is in the transfer frame primary header (see Figure 3-4). The meaning of the application process ID shall be static for the complete mission phase.

3.1.2.2 <u>Packet Sequence Control.</u>

Grouping Flags. The first sub-field of the packet sequence control field is called "grouping flags," and provides for a logical representation of four types of grouping status. Grouping is no longer used per issue 4 of the CCSDS Packet Telemetry Recommendation; however, the 2 bits for the grouping flags remain and are always set to "11" to signify no grouping.

Source Sequence Count. This second sub-field calls for each packet to be numbered in a sequential manner, thus providing a method of checking the order of source application data at the receiving end of the system. It is normally used for accounting purposes to measure the quantity, continuity, and completeness of the data received from the source. The field provides a straight sequential count to modulo 16,384. Longer-term unambiguous ordering (beyond 16,384 packets) may be accomplished by associating the measurement time code contained within the packet with the source sequence count.

3.1.2.3 <u>Packet Length.</u> The last major field of the primary header delimits the boundaries of the packet. It is a count of the number of octets in the packet beginning with the first octet after the 48-bit primary header and ending with the last octet of the packet. The 16-bit field allows packet lengths up to 65,536 octets (not counting the 48-bit primary header). This packet limit was a compromise between the majority of users who produce medium-size packets and the few users who may produce exceptionally long packets. Placing a reasonable limit on packet size helps avoid the flow control problems associated with very long packets, and eliminates the overhead penalty of a larger length field for the great majority of packet producers.

3.1.2.4 <u>Data Field.</u> The remainder of the packet may consist of any data desired, although some suggestions are provided in the application notes, annex B. The total length of all subsequent data should be an even number of octets (a multiple of 16 bits) for efficiency in computer

processing. In addition, Figure 3-2 indicates two possible sub-fields: secondary header and source application data.

Secondary Header. A secondary header may be desirable for providing ancillary data generated by the application process (time, vehicle position/attitude). A secondary header ID field is the first field in the secondary header. The ID field contains two sub-fields. A time code present sub-field provides a bit to indicate the presence of a time code format. A secondary header length sub-field provides the length of the secondary header. If time is included in the secondary header, it must consist of a time code preamble field (P- field), a time specification field (T field), and adhere to either a day or calendar binary segmented or unsegmented binary Time Code Format specified in Appendix C of this standard. The ASCII time code formats are not allowed due to the absence of a preamble identifier for ASCII codes.

Source Data. Following the secondary header, the source data sub-field contains source application data generated by the application process identified in the primary header. For efficiency in computer processing, this sub-field should be a multiple of 16 bits.

Packet Error Control. NOT USED

3.1.3 **<u>Flow Control Mechanisms.</u>** Telecommunications systems are usually constrained by the capacity or the bandwidth of the telecommunications channel/on-board recording device. Flow control becomes crucial when multiple application processes must share the same telecommunications channel if significantly high data rate data is to be recorded. The Digital Data Acquisition and On-Board Recording System must ensure that all sources have proper access to this common resource frequently enough to ensure timely recording as well as to control the need to buffer data while other sources are being serviced. Long source packets may present flow control problems if they monopolize the data channel for unacceptable periods of time while forcing other sources to implement unreasonably large buffering of their data. Several alternative solutions to the problem of flow control are presented below, and are summarized in Annex A-3 of this report.

3.1.3.1 **<u>Virtual Channelization.</u>** One solution to the flow control problem is to assign each source (which generates long packets) its own virtual channel. This is accomplished by inserting these packets into specially identified transfer frames. These dedicated frames form a virtual channel and may be interleaved with other frames containing data from other applications. Detailed discussion of virtual channelization occurs in section 3.1.4.

3.1.3.2 **<u>Source-Internal Segmentation: Source Packet (Version 1).</u>** [No longer used per issue 4 of CCSDS Packet Telemetry Recommendation]

3.1.3.3 **<u>Spacecraft Segmentation: Source Packet (Version 1).</u>** [No longer used per issue 4 of CCSDS Packet Telemetry Recommendation]

3.1.3.4 **<u>Spacecraft Segmentation: Telemetry Segment (Version 2).</u>** [No longer used per issue 4 of CCSDS Packet Telemetry Recommendation]

3.1.4 Transfer Frame. The source packet data structures described in the previous sections are embedded within a data transfer structure, the transfer frame, which provides reliable, error-controlled transfer to and onto the recording media. The transfer frame has a fixed length for a given mission or vehicle. The attributes of the transfer frame and its supporting rationale are presented in the discussion of the transfer frame format. Figure 3-3 illustrates the transfer frame format.

The transfer frame data structure of Figure 3-3 is identical to the transfer frame data structure of the CCSDS Packet Telemetry Recommendation (reference [1]).

3.1.4.1 Synchronization Marker. Attached to the beginning of the transfer frame primary header is a 32-bit frame synchronization marker used by the ground station to acquire synchronization with the frame boundaries during playback. The particular bit pattern is found in Appendix B of this standard. All transfer frames in a single physical data channel in a given mission must be of constant length.

3.1.4.2 Frame Identification. The first major field of the transfer frame primary header is the frame identification field.

Version Number. Only one version of the transfer frame has been defined, although this 2-bit field allows growth to four. The "version" refers to the frame structuring principles that are described in this section.

Vehicle ID. The vehicle identification field provides for positive identification of the vehicle that generated the transfer frame. The 10 bits assigned to vehicle identification allows up to 1024 separate positive IDs.

The CCSDS Packet Telemetry Recommendation defines this field as "spacecraft ID." The CCSDS Packet Telemetry Recommendation also requires the CCSDS Secretariat to assign spacecraft identifiers. This standard changes the name to "vehicle ID" and leaves the authority for assigning vehicle IDs to the individual projects/ranges.

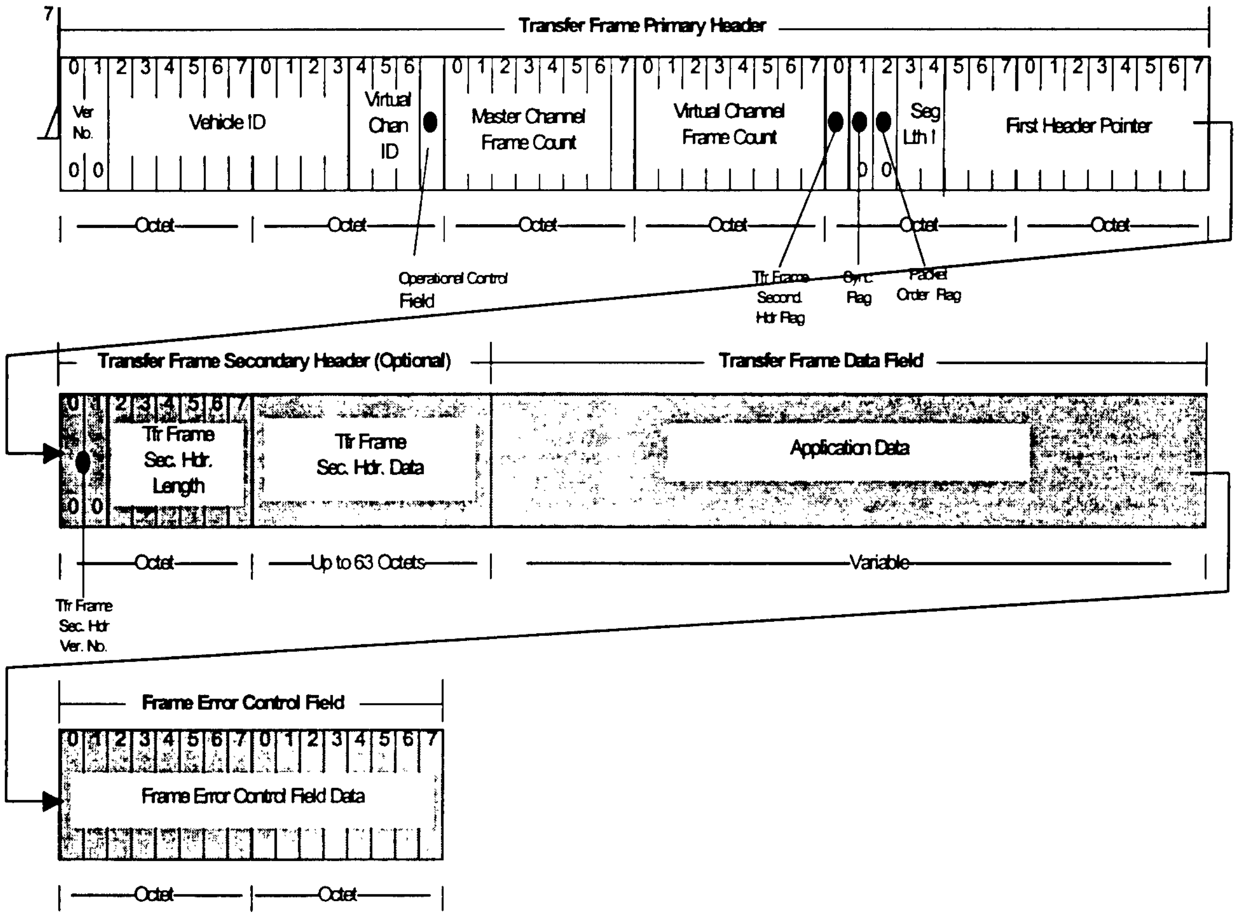

Figure 3-3. Transfer frame format.

Virtual Channel ID. This three-bit sub-field allows up to eight virtual channels to be run concurrently on a particular physical data channel. Frames from different virtual channels are multiplexed together on the recording medium, and, with this identifier in each frame, can be easily split apart during playback on the ground. Virtual channels can be used for a variety of purposes such as flow control to prevent long packets from "hogging" the channel; selecting out different types of data for stream splitting (e.g., when low-rate engineering data must be split out from multiplexed high-rate science data) or when different levels of data quality are to be accommodated for different types of data (in which case error protection may be applied to certain virtual channels but not others). Eight virtual channels are considered sufficient to provide adequate flexibility for envisioned future vehicles.

Operational Control Field Flag. The last bit of the frame identification field, when set to "one," signals the presence of the 32-bit operational control field that is contained within the frame trailer. The RCC Digital Data Acquisition and On-Board Recording Standard does not specify the use of the operational control field, and the operational control field flag is always set to "zero."

For the CCSDS Packet Telemetry Recommendation, the information in this field is defined to provide a standardized spacecraft reporting mechanism for spacecraft telecommanding (uplink) which is not used in this standard.

3.1.4.3 **<u>Master Channel and Virtual Channel Frame Count.</u>** The next two fields provide a running count of the number of frames transmitted. These counters provide a degree of data accountability (for short duration data loss), the ambiguity level being defined by the field lengths.

Master Channel Frame Count. This 8-bit field provides sequential count (modulo 256) of the number of frames transmitted by a single physical data channel. The counter is long enough to provide a reasonable probability of detecting a discontinuity, in a sequence of frames, when the physical channel is briefly interrupted. If such a discontinuity does occur, the virtual channel accounting process can provide a greater probability of detecting the number of missing frames.

Virtual Channel Frame Count. The following 8-bit field provides accountability for each of the eight independent virtual channels. This field is used with the virtual channel ID sub-field to provide accountability via a sequential count (modulo 256). The rationale for the counter ambiguity level is the same as for the master channel frame counter. If only one virtual channel is incorporated for a given mission, both the virtual channel frame counter and the master channel frame counter must increment once per generated transfer frame (i.e., the two fields should not be concatenated into a master frame counter).

3.1.4.4 **<u>Frame Data Field Status.</u>** The frame data field status provides control information that allows the recorder playback end to extract and reconstitute packets and/or segments.

Secondary Header Flag. The first sub-field indicates the presence or absence of the optional secondary header. If its presence is so indicated, the secondary header must appear in every frame transmitted through a physical data channel, and its length must also be fixed. The rationale for this requirement is provided later in the discussion (section 3.1.4.5) about the secondary header.

Synchronization Flag. This flag indicates whether or not the packet or segment data units are inserted into the transfer frame data field on octet boundaries. If they are, then they are said to be "synchronously inserted" (packet octet boundaries align with frame octet boundaries) and the extraction technique (pointing to specific octet) is valid. If the flag indicates "asynchronous" data insertion (i.e., unstructured (non-packetized) data contents or packets inserted without regard to octet boundaries), the transfer frame layer at the recorder playback end will not be able to reconstitute the original data sets without additional knowledge.

Packet Order Flag. This flag indicates whether the sequence count order of the contained packet is increasing (forward) or decreasing (reverse). This has important implications when tape recorded data are played back opposite to their recorded direction. When this is the case, the vehicle electronics re-justifies the bit direction of each packet so each packet individually flows in the forward direction and its header can be read to allow proper packet extraction from the

transfer frame. Even though the playback packets appear individually to flow the same as the rest of the data, the sequence of packets will be running backwards in time as indicated by the decreasing sequence counter. A discussion of various options for handling tape recorded data is contained in annex B.

Segment Length ID. [No longer used per, CCSDS Packet Telemetry Recommendation, issue 4, November 1995.]

First Header Pointer. The first header pointer sub-field points directly to the location of the starting octet of the first packet header structure within the frame data field. It counts from the end of the primary header (secondary header if present) and effectively delimits the beginning of the first packet. The packet length field, in turn, delimits the beginning of the next packet, and so on. Since the pointer counts octets, this feature works only when the headers are aligned with octet boundaries, i.e., when the packet data are synchronously inserted (data field synchronization flag set to zero). The eleven bits allocated to the pointer allow for a count to 2048 octets, which exceeds the count required to point to an octet at the end of the data field. Two special pointer values are reserved. These denote:

(1) No packet header is contained in this frame, but there is valid data, or
(2) No valid data is contained in this frame (idle channel).

3.1.4.5 Frame Secondary Header (Optional). An optional secondary header is provided for users who desire a means for inserting time or for deterministically inserting real-time data (e.g., Time-Division-Multiplexed data). When the secondary header presence is indicated by the secondary header flag, its length must be of a fixed value and must appear in every frame transmitted through a physical channel. Given the requirement for fixed transfer frame length, a fixed secondary header length simplifies data processing and packet extraction at the recording playback end.

Secondary Header ID. The first part of the secondary header has two sub-fields. The first is the secondary header version number, a 2-bit field allowing four versions (or structuring rules). Only version 1 (represented by 00) is currently defined. This provides for a reasonable future growth capability.

The second sub-field, secondary header length, indicates what length has been selected for the secondary header. This 6-bit sub-field provides a binary count of the total number of octets contained within the entire transfer frame secondary header (including the ID field itself, which is one octet in length). This limits the total secondary header length to 64 octets (512 bits) which is considered adequate for currently understood applications.

Secondary Header Data. This sub-field contains up to 504 bits of user specified data. Specifically, if time is to be included in the secondary header data field, it must conform to the Time Code Formats of appendix C.

3.1.4.6 Transfer Frame Data Field. The transfer frame data field contains an integral number of octets of data (source packets) to be recorded by the on-board recording device. The maximum length of this field depends on which optional fields are implemented.

3.1.4.7 Transfer Frame Trailer. A transfer frame trailer provides for frame error detection.

Operational Control Field. NOT USED IN THIS STANDARD.

The CCSDS Packet Telemetry Recommendation uses the Operational Control Field to uplink Telecommand information.

Frame Error Control Field. This field occupies the two trailing octets of the transfer frame. Its presence is mandatory and must appear in all transfer frames. It provides the capability for detecting errors that have been introduced into the frame during the data handling processes.

A Cyclic Redundancy Code (CRC) has been selected for this purpose because of its effectiveness and simplicity, and is defined and specified in section 5.5.1 of this standard. Parity is generated over the entire transfer frame (less the final 16 bits), and the 16 bits of parity checks are then appended to complete the frame. It should be noted that in the 1984 issue of the CCSDS Packet Telemetry Recommendation, the frame was defined to include the "attached sync marker." In the 1987 issue, the frame definition was changed to exclude the marker, but it was still considered to be "attached." To maintain compatibility with already-built systems, it was necessary to allow for two options over which the CRC is applied, that is; it may include the sync marker or it may exclude it. Since the marker pattern is always known, the preferred choice is to omit the marker when encoding. If the frame error control field is not utilized, it is recommended to fill the field with all "ones" or "zeroes."

3.2 TELEMETRY CHANNEL CODING

The channel coding layer is not specified in this standard.

The CCSDS Packet Telemetry Recommendation includes a Channel Coding Recommendation as an integral part of the recommendation. Because of the over-the-air transmission issues, the Channel Coding Recommendation is appropriate. However, for this standard, data errors and/or loss on recording are significantly less than experienced in over-the-air transmission of telemetry signals. Therefore, the specific coding implementation is left to the discretion of the implementor depending on the characteristics of the physical recording medium. (Future versions of this standard may specify a coding layer standard).

ANNEX A-1

GLOSSARY OF TERMINOLOGY

fill bit(s) - Additional bit(s) appended to enable a "data entity" to exactly fit an integer number of octets.

octet - An 8-bit word consisting of eight contiguous bits.

packet - An efficient application-oriented protocol data unit that facilitates the transfer of source data to users located in space or on Earth.

physical channel - The single bit stream that is recorded onto the physical media. It includes all multiplexed transfer frames as well as any implementor unique coding algorithms.

protocol - A set of procedures and their enabling format conventions that define the orderly exchange of information between entities within a given layer of the TM System.

reliable - Meets the quality, quantity, continuity, and completeness criteria that are specified by the TM System.

Digital Data Acquisition and On-Board Recording System - The end-to-end system of layered data handling services that exist to enable a vehicle to send measurement information, in an error-controlled environment, to on-board recorders and then played back through data processing equipment on Earth.

transfer frame - A communication-oriented protocol data unit that facilitates the transfer of application-oriented protocol data units through the space-to-ground link.

user - A human or machine-intelligent process which directs and analyzes the progress of a mission.

virtual channel - A given sequence of transfer frames, which are assigned a common identification code (in the transfer frame header), enabling all transfer frames that are members of that sequence to be uniquely identified. It allows a technique for multiple source application processes to share the finite capacity of the physical channel (i.e., through multiplexing).

ANNEX A-2

APPLICATION NOTES

Purpose:

The Digital Data Acquisition and On-Board Recording Standard provides extensive flexibility in formatting a wide variety of data types for the purpose of recording them. While the basic formats are standardized, the implementor still must make choices when using this standard to format data for recording.

The use of this standard alone does not imply interoperability between the processes that format the data for recording and those that subsequently read the recording and interpret the data.

These application notes have been developed to ensure that a process can be implemented to extract application data from the Digital Data Acquisition and On-Board Recording Format and to interpret it as it was intended to be interpreted. The specific purpose of the application notes, therefore, is to guide the implementors in making commonly accepted choices when applying Digital Data Acquisition and On-Board Recording Formats to record the most frequently recorded types of data.

Status:

These application notes are subject to periodic revision by the RCC Telemetry Group and are to be considered a work in progress.

The goal of these reviews is to eventually achieve a single application layer standard for each data type instead of the dual standards that now exist for some data types.

Annex Organization:

The application notes are organized into sections each of which describes guidelines related to a particular type of application data stream presented to the entity which implements this standard. The first section gives general guidelines, while the remaining sections contain guidelines of particular relevance to specific types of data to be recorded or particular layer processes.

1. GENERAL GUIDELINES

The following guidelines may be of interest to implementors of the standard irrespective of the type of data being recorded.

1.1. SELECTION OF TRANSFER FRAME LENGTH

The standard specifies that the length of a transfer frame is fixed for a mission. When the recording capacity during a mission is of concern, the transfer frame length should be made as large as possible to reduce the amount of recording space used for transfer frame primary header and transfer frame secondary header octets. In contrast, to meet operational requirements, the maximum transfer frame length may be limited by multiplexing needs, data criticality, or buffer size.

The maximum length of a transfer frame is limited by appendix B to 8920 bits.

1.2 TRANSFER FRAME BUFFERING CONSIDERATIONS

For channels where data may not arrive for long periods of time, and when there is no data to be recorded, a transfer frame filled with idle data may be forwarded to mark time on the virtual channel. This will allow the playback mechanism to find at least one transfer frame from each virtual channel within a short sequence on the recording medium. Section 1.3 of this annex provides guidelines for forwarding partially filled or empty transfer frames.

1.3 FORWARDING OF PARTIALLY FILLED TRANSFER FRAMES

An implementor may choose to have transfer frames forwarded before they are completely filled. For example, there may be a need to forward transfer frames at a constant periodic rate, whether or not there is sufficient data to fill them, to maintain synchrony with the recording equipment, to facilitate playback as noted in section 1.2 of this annex, or to pass user defined information contained in the transfer frame secondary header. Also, the implementor may want to forward the transfer frame at a time increment rollover even if the frame is not completely filled.

When there are not sufficient source packets to fill a transfer frame and a transfer frame must be forwarded on a particular virtual channel, there are two methods available. The first method is applicable if there is already a transfer frame under construction that is partially filled with one or more source packets. For this case, the transfer frame data field may be filled with one or more idle packets so as to completely fill the transfer frame and thereby cause it to be forwarded for recording. If more than 7 octets remain in the partially filled transfer frame, an idle packet should be inserted. If 7 or fewer octets remain in the partially filled transfer frame, fill the remaining bits in the transfer frame with idle data. A second method is applicable if no transfer frame is being filled with source packets when a transfer frame must be forwarded. In this case, it is left as an implementation option whether to forward a complete transfer frame containing

only idle data or containing one or more idle packets that completely fill the transfer frame data field.

1.4 OCTET ALIGNMENT

All data placed into the source data field of a source packet should end on an octet boundary. If there are not sufficient application data bits in the source data field of a source packet to make up an integral number of octets, one to seven fill bits should be appended to the end of the application data to make the source data field an integral number of octets in length (see following sections for details on octet alignment options relative to specific application data types).

1.5 TREATMENT OF TIME VALUES

The time values used in this standard are associated with several different events. Care must be taken in how these time values are used. There are two major associations of time values: time associated with the application data, and time associated with the packaging of the application data using the formats (source packets and transfer frames) prescribed by this standard.

1.5.1 Time Associated with the Application Data. When there is a critical need for associating time with an application data stream, the time values should be contained in the application data stream itself and will be considered part of the application data stream. Time information which is embedded into the application data stream is considered independent of any time information which may be included in the source packet secondary header or the transfer frame secondary header (see section 1.5.2 of this annex). Time information that is embedded into the application data is not covered by this standard.

1.5.2 Time Associated with the Source Packet and Transfer Frames. This standard provides the optional capability to carry source packet time in the packet secondary header time code field (see paragraph 3.2.1.1 of this standard) and/or transfer frame time in the transfer frame secondary header data field (see paragraph 5.2.2 of this standard). When source packet time and/or transfer frame time is included

a.) The source packet time and the transfer frame time are considered to be independent of one another.

b.) The CCSDS Time Code Recommendation, Level 1, segmented or unsegmented binary time code formats, shall be used to represent time information (i.e., the time values shall be represented completely and unambiguously). (See reference [3] and appendix C.) ASCII time code formats are not allowed, because preamble identification codes do not exist.

c.) The time code formats shall consist of a Preamble field (P-field) and a Time field (T-field).

d.) The length of the time code field shall be defined by the implementor and resolution shall be indicated in the P-field. (See reference [3] and appendix C.)

e.) The time value shall represent the arrival time of the first bit of data in the source packet data field and/or transfer frame data field.

1.6 SOURCE PACKET TIME

The implementor is free to include source packet time and may use whatever time value precision is sufficient to ensure the efficient processing of source packets (see appendix C).

The method of inserting source packet time codes into the packet secondary header is illustrated in Figure 1-1. For this example, assume that the following time information (all values in decimal) is required in the packet secondary header of a particular source packet using calendar segmented time codes with the day of year calendar variation (see appendix C):

- Day of year (DOY)=245
- Hour (H)=13
- Minute (M)=05
- Second (S)=49
- Second x 10^{-2} (S^{-2})=27

Packet Secondary Header

Packet Secondary Header Identification Field

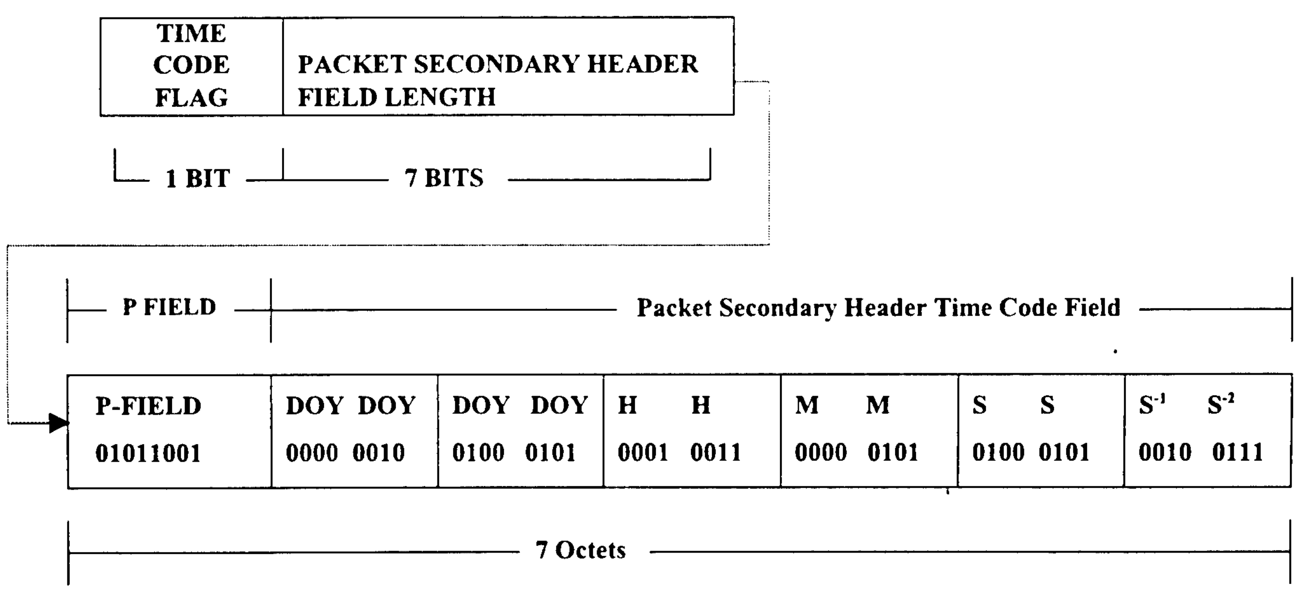

Figure 1-1. Insertion of time codes into packet secondary header.

2. PULSE CODE MODULATION (PCM) APPLICATION DATA

This section pertains to PCM application data structures that are synchronous, fixed length, frame-oriented and generally contain application data from a number of independent sources. Currently the only approved PCM formats are defined in IRIG 106 (Reference [7]).

2.1 APPLICATION LAYER CONSIDERATIONS

Where there is a requirement to time-tag application data with a high degree of accuracy, the implementor should include application data time as one input to the PCM frame in accordance with the format prescribed in Chapter 4 of reference [7]. The reason for requiring application data time to be embedded into the PCM data stream is that there may be a relatively large difference between the time of origin of the application data and the time at which the application data is finally placed into a source packet. The source packet time represents the time at which the source packet was created.

2.2 PACKETIZATION LAYER CONSIDERATIONS

The source packet is allowed to be of variable length depending on the needs of the application process(es). However, with synchronous data, it is considered prudent to use fixed length source packets. The choice of the length of the source packet may be selected freely upon consideration of latency, buffer size requirements, efficiency, and characteristics of the physical media of the recording.

There are two basic approaches for placing PCM application data into source packets - frame aligned, and frame not aligned. These approaches are described in sections 2.2.1 and 2.2.2 of this annex, respectively.

2.2.1 Frame Aligned with Start of Source Data Field. A straightforward approach for placing PCM application data into source packets is to place an integral number of minor frames into a single source packet. Under this approach, the beginning of a minor frame of data is aligned with the beginning of the source data field. There are two acceptable methods for using this approach. One is to place a single complete minor frame into a single source packet. The second method is to place more than one complete minor frame of application data into a source packet.

> *In either case, the implementor may choose to provide for octet alignment on a PCM word basis for all PCM words in the PCM minor frame. If octet alignment is on a PCM word basis, each individual PCM word must either naturally occupy an integral number of octets, or fill bits will be included at the end of each PCM word (LSBs per Figure 1.1 of this standard). If octet aligned on a PCM minor frame basis, only the last PCM minor frame in the packet must end on an integral number of octets. This may require 1-7 fill bits included at the end of the last PCM minor frame (LSBs per Figure 1.1 of this standard)*

a.) Single Minor Frame Per Source Packet - By convention, use of this method for packaging a single minor frame in a source packet implies that the first bit of a minor frame will be found at the first bit position of the source data field. Insertion of the PCM frame shall either be contiguous bits of the PCM stream or octet alignment on a PCM word basis (see section 2.2.1 for PCM word octet alignment). An example of PCM application data implementation with one minor frame per source packet is shown in Figure 2-1.

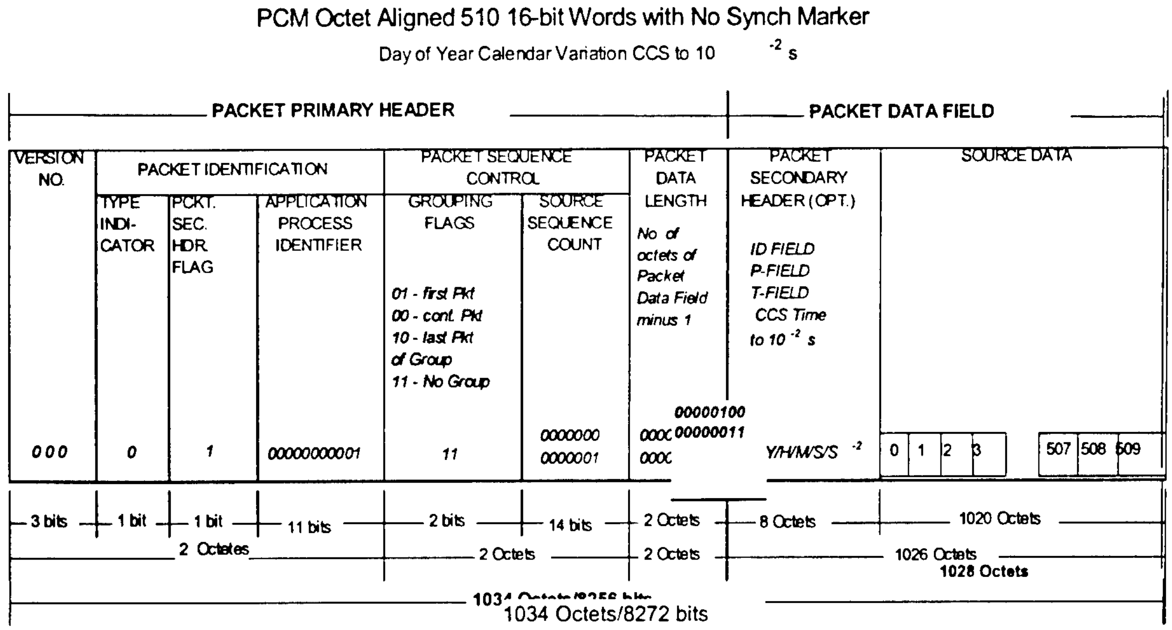

Figure 2-1. PCM, octet aligned with one frame per packet.

b.) Multiple Frames Per Source Packet - By convention under this method, when placing more than one PCM minor frame into a source packet, the first bit of the first PCM minor frame synch marker is placed into the first bit position of the source data field. The first bit of the second PCM minor frame synch marker immediately follows the last bit of the first PCM minor frame, and so on for the remaining PCM minor frames placed into that source packet. An example of a PCM application data implementation with two minor frames per source packet is shown in Figure 2-2.

2.2.2 <u>Frame Not Aligned with Start of Source Data Field.</u> An implementor may choose not to align the PCM minor frame with the start of the source data field of a source packet. Two methods exist. The first method involves stuffing bits of the PCM bit stream, including synch marker, into contiguous bit positions in the entire source data field. The source data field shall be an integral number of octets in length. The source packet length for the PCM bit stream shall be fixed for the duration of the mission. The second method entails aligning each PCM word with the source data field octets (see section 2.2.1 for a description of octet alignment on a PCM word basis). The PCM frames are not aligned with the source data field boundaries.

3. MIL-STD-1553 A/B APPLICATION DATA

MIL-STD-1553 defines the format for messages consisting of command, status, error, and data words. When MIL-STD-1553 messages are recorded using this standard, the 24-bit word structure defined in Chapter 8 of reference [7] for telemetry output shall apply with the exceptions noted in the following sections. In this Annex, MIL-STD-1553 words represented in the Chapter 8 24-bit structure are referred to as "IRIG 106-formatted 1553 words" and MIL-STD-1553 messages represented in Chapter 8 format shall be called "IRIG 106-formatted 1553

messages." The primary departure of this standard from the Chapter 8 conventions is that there is no need to create a composite output PCM frame structure as defined in section 8.6 of reference [7] for transporting the IRIG 106-formatted 1553 messages because under this standard the source packet effectively replaces the PCM frame. (Note: IRIG 106-formatted 1553 messages in a composite output PCM frame structure per Chapter 8 of reference [7] are treated as PCM application data which is described in section 2 of this annex).

To release the first issue of this standard, only the MIL-STD-1553 A/B application data format contained in IRIG 106 (reference [7]) was considered. Future issues of this standard will consider other more efficient techniques for packetization of MIL-STD-1553 A/B application data.

3.1 APPLICATION LAYER CONSIDERATIONS

MIL-STD-1553 messages shall be formatted and presented to the entity performing packetization layer functions, i.e., creating source packets, as a sequence of IRIG 106-formatted 1553 words as defined in section 8.4 of reference [7]. In addition, where time values need to be inserted into the IRIG-formatted 1553 message, the time words will include embedded high-order, low-order, and microsecond time words formatted according to section 8.5 of reference [7].

3.2 PACKETIZATION LAYER CONSIDERATIONS

For the purposes of placing data into source packets, the key difference between the packetization of PCM data described in Section 2 of this annex and the packetization of IRIG 106-formatted 1553 messages is that the latter are of variable length. When recording IRIG 106-formatted 1553 messages, the maximum source packet length may be freely selected upon consideration of latency, buffer size requirements, efficiency, and characteristics of the physical media of the recording. Because a source packet can vary in length, the implementor may choose to forward a source packet even if it has not been filled to its maximum length.

3.2.1 <u>Insertion of Time Values into IRIG 106-Formatted 1553 Messages.</u> The implementor shall insert time values into IRIG 106-formatted 1553 messages. When placing time values into an IRIG 106-formatted 1553 message, the technique described in section 8.5 of reference [7] applies. Use of this technique requires the placement of all three time words, i.e., high-order time, low-order time, and microsecond time, into each IRIG 106-formatted 1553 message contained in the source packet. This method is illustrated in Figure 3-1 which shows only the contents of the source data field of a source packet.

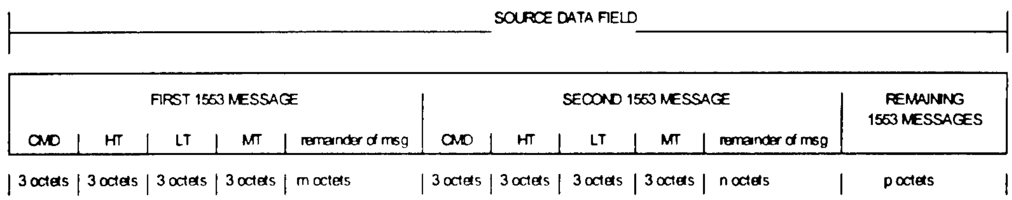

Figure 3-1 Source data field containing IRIG 106-formatted 1553 messages with complete time values in each message.

3.2.2 <u>Source Data Field Structure.</u> Figure 3-2 describes the general format for source packets containing IRIG 106-formatted 1553 messages. By convention, the contents of the source data field shall contain an integral number of IRIG 106-formatted 1553 words, i.e., an IRIG 106-formatted word cannot be split between two source packets. In the example of Figure 3-2, the source data field is 900 octets in length which means that it can hold up to 300 words of IRIG 106-formatted 1553 messages where each word is 3 octets (24 bits) in length.

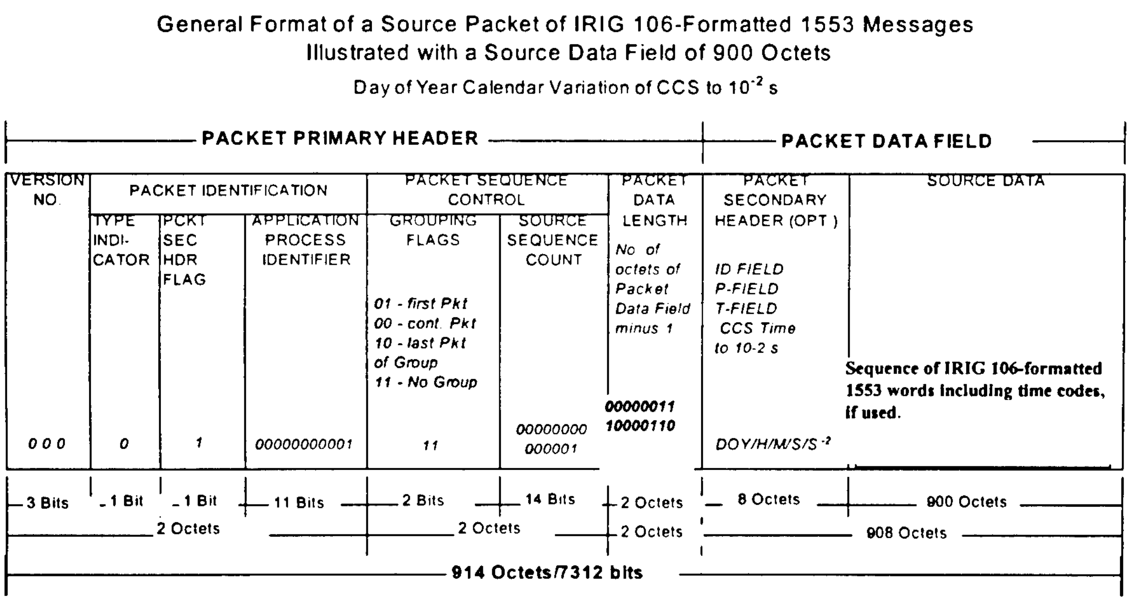

Figure 3-2. Example of the general format for placing IRIG 106-formatted 1553 messages into source packets.

The implementor does not need to align messages to source data field boundaries. The stream of messages is actually viewed more simply as a stream of IRIG 106 formatted words. The implementor may choose to pack as many whole words into the source data field as is convenient. Thus, a single IRIG 106 formatted message may span two source packets. This

method of recording IRIG 106 formatted 1553 messages is illustrated in Figure 3-3 which shows the tail of one message, a complete second message, and the beginning of a third message. The bus ID's shall remain constant for unique application IDs.

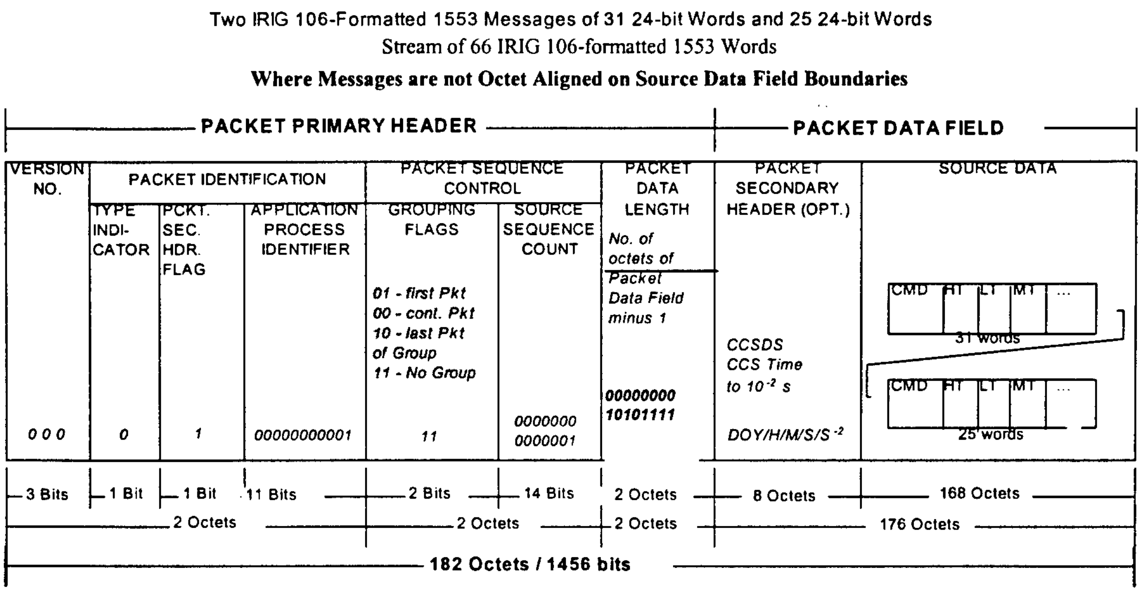

Figure 3-3. IRIG 106-formatted 1553 messages not aligned with source packet illustrated with 240 IRIG 106-formatted 1553 words.

4. PACKETIZATION OF OTHER APPLICATION DATA TYPES

This standard can be used for any other bit stream data, parallel data, asynchronous data, communications data, etc., that can be identified using an application ID and can be placed into source packets as contiguous bits as the data arrives without special formatting at the packetization layer. It is left to the implementor to determine how this data is to be placed into source packets and how to recreate the higher order structure of the original bit stream.

To release the first issue of this standard, no other application data types were specifically considered. Future issues of this standard may provide application notes for specific techniques for packetization of other application data types.

5. DIGITAL VOICE APPLICATION TELEMETRY DATA

Digital voice will normally be an input to a PCM multiplexed data stream. It therefore does not require any special application guidelines other than those described for frame-oriented multiplexed application data.

6. RCC DIGITAL DATA ACQUISITION AND ON-BOARD RECORDING STANDARD DATA

In the future, application data systems may be designed to conform to this standard. In that case, the application process will provide integral source packets of data to the Digital Data Acquisition and On-Board Recording System. These data will be directly inserted into transfer frames for on-board recording.

7. CODING LAYER INTERFACE CONSIDERATIONS

This standard does not encompass the coding layer and therefore does not prescribe how to identify the start of transfer frames. It also does not specify how channel coding is to be applied except as described in appendix B.

7.1 USE OF AN ATTACHED SYNCH MARKER

The use of an Attached Synch Marker (ASM) to indicate the start of a transfer frame is required. It shall conform to the specification as described in appendix B, which provides the appropriate excerpted sections of reference [2].

7.2 CHANNEL CODING

Although not specified in this standard, channel coding for purposes of error detection and/or correction must be integrated into the recording device and be transparently removed during playback. (Future versions of this standard may specify a coding layer standard.)

8. HEADER RECORDS

A header record may be written to a recording device to describe the contents of the recording. A future addition to the application notes will specify the use of the Telemetry Attributes Transfer Standard (TMATS) described in Chapter 9 of reference [7] to convey header information. There is no explicit definition of the contents of header records at this time.

ANNEX A-3

TRANSFER FRAME ERROR DETECTION ENCODING/DECODING GUIDELINE

A3-1 CODING FOR ERROR DETECTION IN TRANSFER FRAMES

This annex describes the error detection encoding/decoding procedure that is recommended for transfer frame coding.

The code specifies the same generator polynomial used by HDLC (ISO), ADCCP (ANSI), V.41 (CCITT), etc. It has the following capabilities when applied to an encoded block of less than 32,768 (2^{15}) bits:

(1) All error sequences composed of an odd number of bit errors are detected.

(2) All error sequences containing at most two bit errors anywhere in the encoded block will be detected.

(3) If a random error sequence containing an even number of bit errors (greater than or equal to 4) occurs within the block, the probability that the error will be undetected is approximately 2^{-15} (or approximately 3×10^{-5}).

(4) All single error bursts spanning 16 bits or less will be detected provided no other errors occur within the block.

A3-1.1 Encoding Procedure

The encoding procedure accepts an (n-16)-bit data block and generates a systematic binary (n,n-16) block code by appending a 16-bit Frame Check Sequence (FCS) as the final 16 bits of the codeblock. This FCS is inserted into the frame error control word of the transfer frame trailer. The equation for the FCS is:

$$FCS = [X^{16} \cdot M(X) \quad X^{(n-16)} \cdot L(X)] \text{ modulo } G(X)$$

where

M(X) is the (n-16)-bit message to be encoded expressed as a polynomial with binary coefficients

L(X) is the presetting polynomial given by:

$$L(X) = \sum_{i=0}^{15} x^{**}I \text{ (all “1” polynomial of order 15)}$$

G(X) is the generating polynomial given by:

$$G(X) = X^{16} + X^{12} + X^{5} + 1$$

n is the number of bits in the encoded message

$\oplus$ is the modulo 2 addition operator (Exclusive OR)

Note that the encoding procedure differs from that of a conventional cyclic block encoding operation in that:

The $X(n-16) \cdot L(X)$ term has the effect of presetting the shift register to an all "1" state prior to encoding.

A3-1.2 Decoding Procedure

The error detection syndrome, S(X), is given by

$$S(X) = [X^{16} \cdot C^*(X) \quad X^{n} \cdot L(X)] \text{ modulo } G(X)$$

where C*(X) is the received block in polynomial form and S(X) is the syndrome polynomial which will be zero if no error is detected and non-zero if an error is detected.

A3-2 POSSIBLE IMPLEMENTATION

A possible implementation of the above-defined encoding/decoding procedure is described below.

A3-2.1 Encoding

Figure A3-1 shows an arrangement for encoding using the shift register. To encode, the storage stages are set to "one," gates A and B are enabled (closed), gate C is inhibited (open), and (n-16) message bits are clocked into the input. They will appear simultaneously at the output. After the bits have been entered, the output of gate A is clamped to "zero," gate B is inhibited, gate C is enabled, and the register is clocked a further 16 counts. During these counts the required check bits will appear in succession at the output.

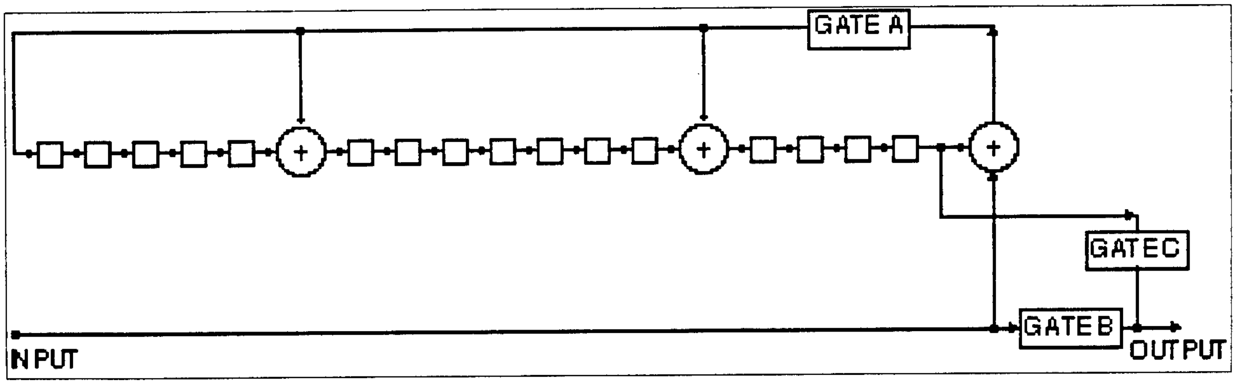

Figure A3-1. Encoder.

A3-2.2 Decoding

Figure A3-2 shows an arrangement for decoding using the shift register. To decode, the storage stages are set to "one" and gate B is enabled. The received n-bits [the (n-16) message bits plus the 16 bits of the FCS] are then clocked into the input. After (n-16) counts, gate B is inhibited. The 16 check bits are then clocked into the input, and the contents of the storage stages are then examined. For an error-free block, the contents will be zero. A non-zero content indicates an erroneous block.

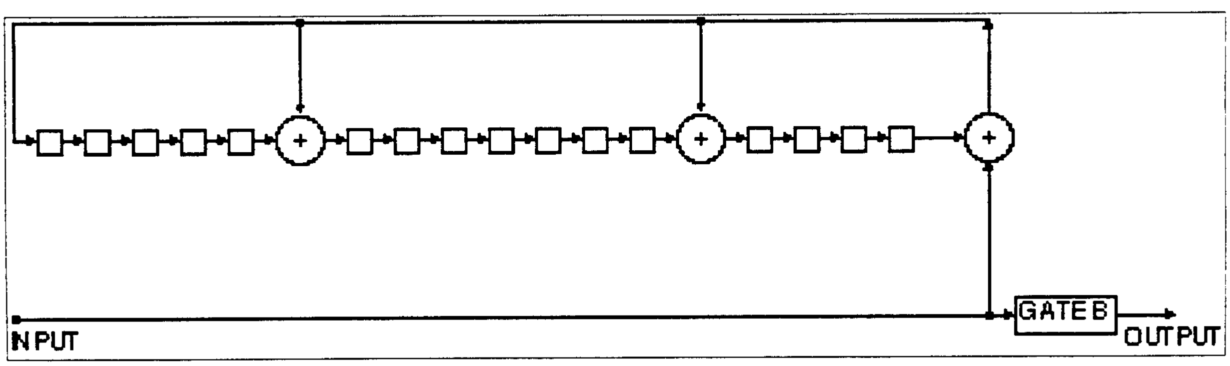

Figure A3-2. Decoder.

APPENDIX B

CHANNEL CODING

This appendix is excerpted from the CCSDS "Channel Coding Recommendation," reference [2]. The paragraph numbers of reference [2] are retained for ease of correlation. Any paragraph numbers not *specifically* identified in this appendix are not being adopted by this standard.

4.2 SPECIFICATION

The Transfer Frame is defined by paragraph 2.3 of this Standard. It has a maximum length of 8920 bits, not including the 32-bit Attached Sync Marker.

5.2 ASM BIT PATTERN

The ASM for the telemetry channel data stream shall consist of a 32-bit (4-octet) marker with a pattern as follows:

0001 1010 1100 1111 1111 1100 0001 1101

FIRST TRANSMITTED BIT (Bit 0) LAST TRANSMITTED BIT (Bit 31)

The pattern is represented in hexadecimal notation as:

1ACFFC1D

The ASM is attached to (i.e., shall immediately precede) the Transfer Frame.

The ASM for one Transfer Frame shall immediately follow the end of the preceding Transfer Frame; i.e., there shall be no intervening bits (data or fill) preceding the ASM

APPENDIX C

TIME CODE FORMATS

This appendix is excerpted from CCSDS "Time Code Formats", reference [2]. The paragraph numbers of reference [2] are retained for ease of correlation. Any paragraph numbers not *specifically* identified in this appendix are not being adopted by this standard.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Purpose

The purpose of this recommendation is to establish a small number of standardized recommended time code formats for use in data interchange applications between organizations of the RCC. This recommendation does not address such timing performance issues as stability, precision, accuracy, etc.

1.2 Scope

Time codes are digital representations of time information. Four standard CCSDS-recommended time codes are described (one unsegmented and three segmented codes) that use the international standard second as the fundamental unit of time. An unsegmented time code is a pure binary count of time units and fractional time units from a starting time called the "epoch." A segmented time code is one in which the count of time units and fractional time units is accumulated in two or more cascaded counters that count the modulo of various bases and start from the epoch.

The four recommended time code formats carry both the time data (in the time specification field, or T-field) and, where applicable, additional information (in the time code preamble field or P-field) that uniquely identifies a specific time code format. The P-field may be either explicit or implicit (refer to paragraph 2.1.1 of this appendix).

1.3 Categorizing of RCC Time Code Formats

In this recommendation, four levels of time code formats can be defined based on the four degrees of interpretability of the code. All time code levels provide for recognizing the boundaries of the time code field and thus that field can be transferred as a block to another location.

Level 1: Complete Unambiguous Interpretation *(This is the only Level currently adopted by this standard).*

Level 1 code formats are fully self-defined and allow absolute time interpretation for the events tagged with the code. Time comparison with other time sources utilizing level 1 codes can thus be made. These codes are the RCC-recommended codes and have the recommended epochs.

1.4 Applicability

This recommendation contains a number of time codes designed for applications involving data interchange in remote data systems. It does not attempt to prescribe which code to use for any

particular application. The rationale behind the design of each code is described in annex B of this appendix and may help the application engineer to select a suitable code. Definition of the timing accuracy underlying a particular time code is not a function of this recommendation, but the responsibility of the authority cognizant of time performance for the applicable system.

1.5 Bit Numbering Convention and Nomenclature

In this document, the following convention is used to identify each bit in an N-bit field. The first bit in the field to be transmitted (i.e., the most left justified when drawing a figure) is defined to be Bit 0; the following bit is defined to be Bit 1 and so on up to Bit N-1. When the field is used to express a binary value (such as a counter), the Most Significant Bit (MSB) shall be the first transmitted bit of the field, i.e., Bit 0.

BIT 0 Bit N-1

In accordance with modern data communications practice, vehicle data fields are often grouped into 8-bit words that conform to the above convention. Throughout this recommendation, the following nomenclature is used to describe this grouping:

2. TIME CODE FORMATS

The time code formats can be represented as a combination of a preamble field (P) and a time specification field (T). The P-field uniquely defines the options, parameters, and encoding structure of the T-field and should be included whenever the recipient of the time code may be uncertain as to the selected code. The T-field and the P-field shall each be an integral number of octets in length.

2.1 Time Code Fields

2.1.1 Preamble Field (P-Field).

The time code preamble field (P-field) may be either explicitly or implicitly conveyed. If it is implicitly conveyed (not present with T-field), the code is not self-identified, and identification must be obtained by other means. As presently defined, the explicit representation of the P-field is limited to one octet whose format is described as follows:

Bit	Interpretation
0	Extension flag
1 – 3	Time code identification
4 – 7	Detail bits for information on the code

The first bit (Bit 0) of the P-field is the extension flag used to indicate that a second octet is included in the P-field for time code identification. Such an expansion may be required to accommodate new time codes or to provide more information (for example, on the clock used). Presently, the value of this bit is "0", indicating that there is not a second octet present. If a second octet is present, its first bit shall be an extension flag with the same definition: "0" implies it is the last octet of the P-field, "1" implies another octet follows.

The detailed specifications of bits 1 to 7 are given in the following paragraphs with the description of each code. The time code identifications (bit 1 - 3) = 000 and 011 are reserved for future application.

2.1.2 Time Field (T-Field).

For each code, the T-field has a basic structure and optional extensions that allow increases in resolution or ambiguity period.

2.2 CCSDS Unsegmented Time Code (CUC)

2.2.1 T-Field.

For the unsegmented binary time codes described herein, the T-field consists of a selected number of contiguous time elements, each element being one octet in length. An element represents the state of 8 consecutive bits of a binary counter, cascaded with the adjacent counters, which rolls over at a modulo of 256.

The basic time unit is the second. The T-field consists of 1 to 4 octets of coarse time (seconds) and 0 to 3 octets of fine time (subseconds). The coarse time code elements are a count of the number of seconds elapsed from the epoch. Four octets of coarse time results in a maximum ambiguity period of approximately 136 years. This allows a time code representation of time through the year 2094 for those which are referenced to the TAI epoch of 1958 January 1.

Zero to three octets of fine code elements result in a resolution of, respectively, 1 second; 2**8 second (about 4 ms); 2**16 second (about 15 ms); or 2**24 second (about 60 ns).

The RCC-recommended epoch is that of 1958 January 1 (TAI).

This time code is not UTC-based, and leap-second corrections do not apply.

2.2.2 P-Field.

Bit 1 - 3 = Time code identification

001 -- 1958 January 1 epoch (Level 1)
010 -- Agency-defined epoch (Level 2)

Bit 4 - 5 = (number of octets of coarse time) - 1 (For the 1958 epoch, bits 4-5 must be "11" to ensure a long enough ambiguity period).

Bit 6 - 7 = (number of octets of fine time)

2.3 CCSDS Day Segmented Time Code (CDS)

2.3.1 <u>T-Field.</u>

For the segmented binary time code described herein, the T-field consists of a selected number of contiguous time segments. Each segment represents the state of a binary counter, cascaded with the adjacent counters, which rolls over at the modulo specified for each counter.

The segmented binary day count code recommendation, designated CDS (CCSDS Day Segmented), is as follows:

			OPTIONAL
	DAY	ms of day	µs of ms
SEGMENT WIDTH (bits)	16 or 24	32	16

Each segment above is a right-adjusted binary counter. The RCC recommended day segment is a continuous counter of days from 1958 January 1 starting with 0, but other agency-defined epochs may be accommodated as a level 2 code.

The microseconds (ms) segment is optional. Since this code is UTC-based, the leap second correction must be made.

2.3.2 <u>P-Field.</u>

Bit 1 - 3 = Time code identification = 100

Bit 4 = epoch identification

0 -- 1958 January 1 epoch (Level 1)
1 -- Agency-defined epoch (Level 2)

Bit 5 = length of day segment

0 -- 16-bit day segment
1 -- 24-bit day segment

Bit 6 - 7 = resolution (number of optional subsecond segments)

00 -- ms (millisecond)
01 -- µs (microsecond)
10 -- reserved for future use
11 -- reserved for future use

2.4 CCSDS Ca;endar Segmented Time Code (CCS)

2.4.1 T-Field.

For the segmented Binary Coded Decimal (BCD) time code described herein, the T-field consists of a variable number of contiguous time segments. Each 8-bit segment represents two decimal digits.

Both CCS time code variations are UTC-based. The leap second correction must be made.

The calendar segmented code recommendations, designated CCS (CCSDS Calendar Segmented time code), are Level 1 time code formats and are as follows:

2.4.1.1 Month of Year/Day of Month Calendar Variation.

							OPTIONAL					
	YR	MO	DOM	h	m	s	10^{-2}s	10^{-4}s	10^{-6}s	10^{-8}s	10^{-10}s	10^{-12}s
SEGMENT WIDTH (bits)	16	8	8	8	8	8	8	8	8	8	8	8

The year AD segment (YR) requires 16 bits for proper representation of the decimal year. All other segments require 8 bits for proper representation. The month (MO) and day of month (DOM) segments are present when the calendar variation flag (bit 4 of the P-field) is set to zero.

2.4.1.2 Day of Year Calendar Variation.

						OPTIONAL					
	YR	DOY	h	m	s	10^{-2}s	10^{-4}s	10^{-6}s	10^{-8}s	10^{-10}s	10^{-12}s
SEGMENT WIDTH (bits)	16	16	8	8	8	8	8	8	8	8	8

This variation of the CCS time code substitutes day of year (DOY) in place of the month (MO) and day of month (DOM) segments. The day of year segment must be 16 bits long (all segments must be multiples of 8 bits). The four most significant bits of this segment are not used and are set to zero. The day of year segment is present when the calendar variation flag (bit 4 of the P-field) is set to a value of one. The year AD segment is 16 bits in length.

2.4.2 <u>P-Field.</u>

Bit 1 - 3	=	Time code identification = 101
Bit 4	=	calendar variation flag 0 -- month of year/day of month variation 1 -- day of year variation
Bit 5-7	=	resolution (number of optional subsecond segments) 000 -- 1 s 001 -- 10-2 s 010 -- 10-4 s 011 -- 10-6 s 100 -- 10-8 s 101 -- 10-10 s 110 -- 10-12 s 111 -- not used

2.5 CCSDS ASCII Calendar Segmented Time Code (ASCII)

2.5.1 <u>T-Field.</u>

The CCSDS ASCII segmented time code is composed of a variable number of ASCII characters forming the T-field. Both ASCII time code variations are UTC-based and leap second corrections must be made. The time represented is intended to match civil time usage. Therefore, the epoch is taken to be the usual Gregorian calendar epoch of 1 AD, and the time is that of the prime meridian. The ASCII time code recommendations are Level 1 time code formats.

2.5.1.1 <u>ASCII Time Code A, Month/Day of Month Calendar Variation.</u>

The format for ASCII Time Code A is as follows:

YYYY-MM-DDThh:mm:ss.d→ dZ

where each character is an ASCII character using one octet with the following meanings:

YYYY	=	Year in four-character subfield with values 0001-9999
MM	=	Month in two-character subfield with values 01-12
DD	=	Day of month in two-character subfield with values 01-28, -29, -30, or -31
"T"	=	Calendar-Time separator
hh	=	Hour in two-character subfield with values 00-23
mm	=	Minute in two-character subfield with values 00-59

ss = Second in two-character subfield with values 00-59 (-58 or -60 during leap seconds)
d→dZ = Decimal fraction of second in one- to n-character subfield where each d has values 0-9
"Z" = time code terminator (optional)

Note that the hyphen (-), colon (:), letter "T" and period (.) are used as specific subfield separators, and that all subfields must include leading zeros.

As many "d" characters to the right of the period as required may be used to obtain the required precision.

An optional terminator consisting of the ASCII character "Z" may be placed at the end of the time code.

EXAMPLE: 1988-01-18T17:20:43.123456Z

2.5.1.2 ASCII Time Code B, Year/Day of Year Calendar Variation.

The format for ASCII Time Code B is as follows:

YYYY-DDDThh:mm:ss.d→ dZ

where each character is an ASCII character using one octet with the following meanings:

YYY = Year in four-character subfield with values 0001-9999
DDD = Day of year in three-character subfield with values 001-365 or -366
"T" = Calendar-Time separator
hh = Hour in two-character subfield with values 00-23
mm = Minute in two-character subfield with values 00-59
ss = Second in two-character subfield with values 00-59 (-58 or -60 during leap seconds)
d→d = Decimal fraction of second in one- to n-character subfield where each d has values 0-9
"Z" = Time code terminator (optional)

Note that the hyphen (-), colon (:), letter "T" and period (.) are used as specific subfield separators, and that all subfields must include leading zeros.

As many "d" characters to the right of the period as required may be used to obtain the required precision.

An optional terminator consisting of the ASCII character "Z" may be placed at the end of the time code.

EXAMPLE: 1988-018T17:20:43.123456Z

2.5.1.3 Subsets of the Complete Time Codes.

When it is desired to use subsets of each of the two ASCII time code format variations described above, the following rules must be observed:

a. The calendar subset (all subfields to the left of the "T") and the time subset (all subfields to the right of the "T") may be used independently as separate calendar or time formats, provided the context in which each subset is used makes its interpretation unambiguous.

b. When calendar or time subsets are used alone, the "T" separator is omitted.

c. Calendar or time subsets may contain all the defined subfields, or may be abbreviated to the span of interest by deleting the unneeded subfields, either on the left or on the right. However, when subfields are deleted on the LEFT, all separators that had delimited the deleted subfields must be retained (except for the "T" which, by rule b, is dropped if the subset is used alone.) When subfields are deleted on the RIGHT, the separators that had delimited the deleted subfields are dropped.

d. Subsets may NOT consist of partial subfields (e.g., must use "ss", not "s"). In particular, consistent use of the complete four-character YYYY subfield is required (e.g., "1989" instead of "89"). Note, however, that each fractional second ("d" character) is considered to be a complete subfield, and so any number of fractional seconds may be used.

e. If calendar and time SUBSETS are then brought together to form a single time code format (joined with the "T" separator) the CALENDAR subset may NOT have been truncated from the RIGHT, and the TIME subset may NOT have been truncated from the LEFT. That is, the format must be integral around the "T."

f. Standardization on the use of these time code formats for purposes OTHER than identifying an instant of calendar or time in UTC (e.g., unconventional use as a counter or tool for measuring arbitrary intervals) is not recommended. It is felt such a specialized application can best be viewed not as a time code format but rather as an engineering measurement format. Any such application of these time code formats is considered beyond the scope of this recommendation.

ANNEX C-1

RANGE OF SEGMENT COUNTERS FOR

SEGMENTED TIME CODES

(THIS ANNEX IS PART OF THE STANDARD)

Purpose:

This Annex specifies the range of the counters defined in the recommended segmented codes.

RANGE OF TIME CODE SEGMENT COUNTERS

Segment Identification	Range of Counter
Microsecond-of-millisecond	0 to 999
Millisecond-of-day	0 to 86,399,999 (0 to 86,400,999 or 86,398,999 when leap second adjustments are introduced)
Second-of-minute	0 to 59 (0 to 60 or 58 when leap second adjustments are introduced)
Minute-of-hour	0 to 59
Hour-of-day	0 to 23
Day	0 to (2 16 - 1), or 0 to (2 24 - 1)
Day-of-month	1 to 31 for month 1,3,5,7,8,10,12 1 to 30 for month 4,6,9,11 1 to 28 for month 2 (1 to 29 for leap years)*
Day-of-year	1 to 365 (366 for leap years)*
Month-of-year	1 to 12
Year	1 to 9999

* Leap year: every year divisible by 4, except for the years divisible by 100 and not divisible by 400.

ANNEX C-2

RATIONALE FOR TIME CODES

(THIS ANNEX **IS NOT** PART OF THE RECOMMENDATION)

Purpose:

This Annex presents the rationale behind the design of each code. It may help the application engineer to select a suitable code.

C-2 1 GENERAL

Instrument data acquired from vehicles have little value unless it is possible to recreate the significant environment of the instrument during the measurement collection phase. Such ancillary data parameters as time, position, velocity, attitude, instrument temperature, concurrent ground truth measurements and many other parameters may be essential for the proper interpretation of the instrument data. Of these ancillary data parameters, the time of the instrument measurements is certainly the most vital parameter. The reasons for this are the following:

(1) In many cases, the instrument analysis can be based, nearly exclusively, on the sampled sensor time series.

(2) Time provides the most efficient and often the only possible linkage between instrument data and externally generated ancillary parameters. Two independent measurement processes, each correlated with time, can then be correlated with each other.

However, the resulting proliferation of slightly different codes is not desirable. The selection of one particular code will depend on the chosen optimization criteria in the given application. For example, Table C-2-1 compares the four recommended codes in terms of the three selection criteria identified by the CCSDS:

- UTC compatible: Permits unambiguous representation of leap seconds
- Computer efficient: Fewer segments improves data handling and processing
- Human readable: Easily readable code corresponding to widely used civil time representation

Table C-2 -1: Applicability of the Criteria

Time Code	S (Segmented) U (Unsegmented)	UTC Compatible	Computer Efficient	Human Readable
CUC	U	No	Yes	No
CDS	S	Yes	Yes	No
CCS	S	Yes	No	Yes
ASCII	S	Yes	No	Yes

C-2.2 SERVICE RELATED TO THE DIFFERENT LEVELS OF TIME CODE FORMATS

The different levels of the time codes have been distinguished by the self-interpretability of the codes.

All time code levels provide for recognizing the boundaries of the time code field and thus can transfer that field, as a block, to another location.

The different services which can be achieved without special arrangements between users of the CCSDS time codes are:

- Absolute time interpretation: time comparison and differencing for events based on separate time sources, with all sources having the same CCSDS-recommended epoch.

- Relative time interpretation: time comparison and differencing for events based on the same time source, with the source having a known, Agency-defined epoch.

- Ordering of events time-tagged from a single source.

Table C-2-2 shows how these three services can be related to the time code levels.

Table C-2-3 shows the different time code format identifications in the P-field, and the associated time code levels.

Table C-2-2: Time Code Services

Level	Absolute Time Interpretation	Relative Time Interpretation	Ordering
CUC Level 1	Y	Y	Y
CDS Level 1	Y	Y	Y
CCS Level 1	Y	Y	Y
ASCII	Y	Y	Y

Table C-2-3: Service Categories of Time Codes

Time Code Name	Format Identification P-field – (Bits 1-3)	Time Code Category
Reserved	0 0 0	—
C U C	0 0 1	Level 1
C U C	0 1 0	Level 2 (**NOT** used in this Standard)
Reserved	0 1 1	—
C D S	1 0 0	Level 1 or 2 (Level 2 **NOT** used by this Standard)
C C S	1 0 1	Level 1
Agency-Defined	1 1 0	Level 3 or 4 (Levels 3 and 4 **NOT** used by this Standard)
ASCII	111	Level 1 or 3 (Level 3 **NOT** used by this Standard)

C-2.3 DISCUSSION OF RECOMMENDED CODES

All the Recommended time code lengths are an integer number of octets. This helps to optimize the computer processing of these codes and allows the use of high level languages.

The range of all segment counters (especially for leap year and leap second) is shown in Annex A.

C-2.3.1 CCSDS Unsegmented (CUC)

The unsegmented binary time code is particularly suited to computer applications which involve arithmetic computation of time differences. Since the unsegmented format is a representation of the state of consecutive bits of a binary counter (i.e., a continuous function with no discontinuities), arithmetic operations can be carried out directly.

The code allows for both absolute time (TAI scale) and time measured relative to an Agency-defined epoch. Various allowed truncations of the code make it bit-efficient. The attributes of this code make it suitable for applications such as spacecraft clock measurement.

C-2.3.2 CCSDS Day Segmented (CDS)

Most terrestrial time measurements are made using the UTC time scale. Usually, spacecraft instrument events are ultimately time-tagged with UTC because the events have to be correlated with other phenomena. This time code is based on UTC. Since UTC contains discontinuities at the instant of leap second correction, the unsegmented binary code cannot be used to represent UTC.

The CCSDS Day Segmented code (CDS) consists in its simplest form of two binary counters, one counting days from a defined epoch and the other counting milliseconds of day. The code retains attributes similar to the unsegmented binary code in being oriented toward arithmetic operations by computers. The choice of millisecond unit results in an optimum use of 4 octets (28 bits used) and also provides the resolution necessary for most time computations.

Extended microsecond precision is provided by allowing one optional additional segment. Provision has been made in the P-field to accommodate, in the future, greater resolution. CCSDS recommends the epoch 1958 January 1 (Julian date 2436203.5), the epoch of the TAI time scale. An agency-defined epoch is also allowed (such as 1950 January 1). Note that the difference between the epochs 1958 January 1 and 1950 January 1 is exactly 2922.0 days on the Julian date calendar.

The optional 24-bit length of the day segment is included for special applications such as Astronomy. The CDS is the most "machine friendly" of the UTC codes and is therefore particularly suitable for use in computer-to-computer communication requiring very frequent, very fast automated time interpretation and processing.

C-2.3.3 CCSDS Calendar Segmented (CCS)

In human interactions, UTC is frequently expressed in a segmented form consisting of years, months, days, hours, minutes, seconds and decimal fractions of seconds. UTC is also expressed in a segmented form consisting of years, days of year, hours, minutes, seconds and decimal fractions of seconds. The CCSDS Calendar Segmented (CCS) codes (both variations) are oriented towards representing these segments directly in binary coded decimal (BCD) format for ease of human reading and interpretation.

CCS is useful for applications where all or part of the code is to be frequently interpreted by humans, for example, when frequently converting to character form for display purposes. However, CCS is not as efficient as CDS for arithmetic operations.

C-2.3.4 CCSDS ASCII

While binary or BCD-based time code formats are computer efficient and minimize overhead on uplinked/downlinked data streams, there are many ground-segment applications for which an ASCII character-based time code is more appropriate. For example, when files or data objects are created using text editors or word processors, ASCII character-based time code format representations are necessary. They are also useful in transferring text files between

heterogeneous computing systems, because the ASCII character set is nearly universally used and is interpretable by all popular systems. In addition, direct humanly readable dumps of text files or objects to displays or printers are possible without preprocessing. The penalty for this convenience is inefficiency.

The two ASCII time code variations (A, day of month, and B, day of year) include the most widely used human-readable presentations. Both variations are subsets of ISO 8601.

ANNEX C-3

GLOSSARY OF SELECTED TIME TERMS

(THIS ANNEX IS **NOT** PART OF THE RECOMMENDATION)

Purpose:

This annex presents definitions of a number of time-related terms used in the recommendation or useful in understanding the text of the recommendation.

accuracy - Generally equivalent to systematic uncertainty of a measured value (reference [4] – Report 730).

ambiguity period –The interval between successive recurrences of the same time code.

ASCII - A coded set of alphanumeric and control characters used for information interchange. The coded character set used to form the ASCII time codes defined in section 2.5 is described in detail in International Standard ISO 8859-1 (Reference [7]).

atomic time scale - A time scale based on the periodicity's of atomic or molecular phenomena. (See definition of "Unit of Time" - "International Atomic Time Scale".)

Coordinated Universal Time (UTC): (See Universal Time)

date - Synonymous with "time-scale reading", but usually referred to as calendar.

> Note: The date can be expressed in years, months, days, hours, minutes, seconds and fractions thereof.

epoch - The origin (the beginning) of a time scale.

International Atomic Time (TAI): (See Universal Time)

Julian Date - The Julian day number followed by the fraction of the day elapsed since the preceding noon (12 hours UT).

> Example: The date 1900 January 0.5 UT corresponds to JD = 2415020.0.

Julian Day Number - A number of a specific day from a continuous day count having an initial origin of 12 hours UT on 1 January 4713 BC, Julian Calendar (start of Julian Day zero).

> Example: The day extending from 1900 January 0.5 d UT to 1900 January 1.5 d UT has the number 2 415 020.

Modified Julian Date(MJD) - Julian Date less 2 400 000.5 days.

> Note: Other modifications of the Julian date can be created by using other constants; for example:
>
> (1) The constant 2,436,203.5 days, which occurs on 1958 January 1, gives the origin of TAI, recognized as the epoch of both the CCSDS Unsegmented Code (CUC) and the CCSDS Day Segmented Code (CDS).
>
> (2) The constant 2,440,000.5, which occurs on 1968 May 24.0 gives the origin of the Truncated Julian Date (TJD) time scale used in the NASA PB-5J time code.

Precision - Random uncertainty of a measured value, expressed by the standard deviation or by a multiple of the standard deviation.

Time Code Format - A format used to convey time information.

Note: Any representation of time NOT based on the second as the fundamental unit of time is not considered a time code, but is considered to be an engineering parameter. However, it is not necessary for the second to appear explicitly in the time code; decimal multiples or submultiples (e.g., milliseconds of day) may be used.

Time Interval - The duration between two instants read on the same time scale.

Time Scale - A quantitative reference system for specifying occurrences with respect to time.

Time Scale Reading - The value read on a time scale at a given instant.

Time Scale Unit - The basic time interval in a time scale.

Truncated Julian Date - A four-decimal-digit day count originating at midnight 1968-05-23,24.

Universal Time (UT) - In applications in which an imprecision of a few hundredths of a second cannot be tolerated, it is necessary to specify the form of UT which should be used:

UT0 is the mean solar time of the prime meridian obtained from direct astronomical observation.

UT1 is UT0 corrected for the effects of the Earth's polar motion; it corresponds directly with the angular position of the Earth around its axis of diurnal rotation.

UT2 is UT1 corrected empirically for the effects of a small seasonal fluctuation in the rate of rotation of the Earth.

TAI is the international reference scale of atomic time (TAI), based on the second of the International System of Units (SI), as realized at sea level, and is formed by the Bureau International de l'Heure (BIH) on the basis of clock data supplied by cooperating establishments. It is in the form of a continuous scale, e.g., in days, hours, minutes and seconds from the origin 1958 January 1 (adopted by the CGPM 1971).

UTC is the time scale maintained by the BIH which forms the basis of a coordinated dissemination of standard frequencies and time signals. It corresponds exactly in rate with TAI but differs from it by an integral number of seconds.

The UTC scale is adjusted by the insertion or deletion of seconds (positive or negative leap seconds) to ensure approximate agreement with UT1.

Concise definitions of the above terms and the concepts involved are available in the glossary of the annual publication, The Astronomical Almanac (U.S. Government Printing Office, Washington D.C. and H.M. Stationery Office - London).

ANNEX C-4

CONVERSION BETWEEN TAI AND UTC

(THIS ANNEX **IS NOT** PART OF THE RECOMMENDATION)

Purpose:

This annex provides a conversion formula between TAI time and UTC time expressed in seconds.

In the TAI time scale, the CUC code represents a binary count of the elapsed seconds since the 1958 January 1 epoch. Thus it is ideally suited to computation of the true time difference between widely separated events.

In the UTC time scale, CCS time code is the code normally used for time presentation. Computation of the difference between two UTC times requires knowledge of any intervening leap seconds in order to achieve a true difference.

Since January 1, 1972, the relationship between TAI and UTC has been given by a simple accumulation of leap seconds occurring approximately once per year:

At any instant i: Ti = TAI time
Ui = UTC time expressed in seconds
Ti = Ui + Li

where Li is the accumulated leap second additions between the epoch and the instant i.

The following table contains a reference list of the accumulated leap second additions Li between 1972 and 1990:

Time Period	Li
1972 Jan. 1 - 1972 July 1	10.000 000 0 s
1972 July 1 - 1973 Jan. 1	11.000 000 0 s
1973 Jan. 1 - 1974 Jan. 1	12.000 000 0 s
1974 Jan. 1 - 1975 Jan. 1	13.000 000 0 s
1975 Jan. 1 - 1976 Jan. 1	14.000 000 0 s
1976 Jan. 1 - 1977 Jan. 1	15.000 000 0 s
1977 Jan. 1 - 1978 Jan. 1	16.000 000 0 s
1978 Jan. 1 - 1979 Jan. 1	17.000 000 0 s
1979 Jan. 1 - 1980 Jan. 1	18.000 000 0 s
1980 Jan. 1 - 1981 July 1	19.000 000 0 s
1981 July 1 - 1982 July 1	20.000 000 0 s
1982 July 1 - 1983 July 1	21.000 000 0 s
1983 July 1 - 1985 July 1	22.000 000 0 s
1985 July 1 - 1988 Jan. 1	23.000 000 0 s
1988 Jan. 1 - 1990 Jan. 1	24.000 000 0 s
1990 Jan. 1 -	25.000 000 0 s

NOTE: For other periods, see BIH Annual Report.